

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
22 March 2001 (22.03.2001)

PCT

(10) International Publication Number  
**WO 01/20485 A2**

(51) International Patent Classification<sup>7</sup>: G06F 17/00

(21) International Application Number: PCT/US00/25358

(22) International Filing Date:  
13 September 2000 (13.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/153,516 13 September 1999 (13.09.1999) US

(71) Applicant (*for all designated States except US*):  
WWWHOOSH, INC. [—/US]; 160 Old Maple Street,  
Stoughton, MA 02072 (US).

(72) Inventor; and

(75) Inventor/Applicant (*for US only*): BATTERSBY, Clinton,  
Brenton [—/US]; 64 Laurelwood Drive, Stoughton,  
MA 02072 (US).

(74) Agent: CALL, Charles, G.; 53 Saint Stephen Street,  
Boston, MA 02115 (US).

(81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— Without international search report and to be republished upon receipt of that report.

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

WO 01/20485 A2

(54) Title: METHODS AND APPARATUS FOR STORING, RETRIEVING AND MANIPULATING HYPERTEXT DATA

(57) Abstract: A system for increasing the speed of transmission and processing of hypertext data objects, such as HTML web pages and XML documents, that incorporate external data objects by reference, by repackaging existing HTML and XML content and coherency information for more efficient transport over the Internet using existing HTTP and TCP/IP facilities. At the server side of an HTTP link, the content of a hypertext data object is repackaged and renamed as a compound data object which includes data and/or coherency information for imbedded data objects whose URLs are specified in the hypertext object. On the client side, the received compound object is interpreted to avoid the needless transmission of further HTTP requests when valid copies of referenced entities already exist in the local cache, and to obtain the content of the hypertext data object as well as the content of one or more referenced objects in a single request-response interchange. The Appendix contains a Java code implementation of some of the features disclosed.

## PCT Patent Specification

**Title:**           **Methods and Apparatus for  
Storing, Retrieving and Manipulating Hypertext Data**

**Applicant:**   **WWWhoosh, Inc.  
160 Old Maple Street  
Stoughton, MA 02072  
U.S.A.**

**Inventor:**   **Clinton Brenton Battersby  
64 Laurelwood Dr.  
Stoughton, MA 02072  
U.S.A.**

### Technical Field

This invention relates to hypertext communication systems and more particularly to methods and apparatus for speeding the transmission of data objects which incorporate other data objects by reference.

5

### Background Art

The present invention may be used to advantage to more rapidly transmit the information needed for displaying web pages expressed in Hypertext Markup Language (HTML) by means of the Hypertext Transfer Protocol (HTTP). The principles of the invention may also be employed for displaying or processing other forms of information, such as data expressed in SGML and XML which, like HTML, employ data objects which incorporate information found in separate resources which are specified by an suitable identifier, such as an Internet URL, and retrieved using the a suitable request-response network communications protocol, such as the conventional HTTP / TCP-IP transmission mechanism used by the Internet World Wide Web facility.

15

### HTML and XML

HTML, in its preferred forms, as been defined in specifications which have continued to evolve to meet needs of users and developers. HTML 2.0 was developed under the aegis of the Internet Engineering Task Force (IETF) to codify common practice in late 1994 and is described in RFC 1866 (November, 1995). The efforts of the World Wide Web Consortium's HTML

20

Working Group to codify common practice in 1996 resulted in HTML 3.2 (January 1997). HTML 4.0, the latest version of which is available at the URL <http://www.w3.org/TR/html40>, extends HTML with mechanisms for style sheets, scripting, frames, embedding objects, and other enhancements.

5 HTML's multimedia features allow authors to include images, applets (programs that are automatically downloaded and run on the user's machine), video clips, and other HTML documents in their pages. Commonly, in order to completely render a web page (i.e., to display all of the referenced text and images, as well as to play referenced sound and/or video files), it is necessary for the web browser program to scan the HTML text, identify the references to included  
10 resources that need to be fetched, and issue a sequence of separate requests using the Hypertext Transfer Protocol (HTTP) to obtain a current copy of each additional item of referenced data.

The Extensible Markup Language (XML) is a subset of SGML which was designed to enable generic SGML to be served, received, and processed on the Web in the way that was previously possible with HTML. An XML document, as specified in the World Wide Web  
15 Consortium's Recommendation entitled "Extensible Markup Language (XML) 1.0" (February, 1998), may consist of one or many storage units called *entities*; all of which have *content* and which are typically identified by *name*. Each XML document has one entity called the document entity, which serves as the starting point for the XML processor and may contain the whole document. The XML specification permits an XML document to refer to one or more external  
20 entities by an appropriate identifier (URI) so that the content of the external entity referred to may be incorporated into the XML document. Thus, like an HTML web page, the validation or processing of an XML document may require multiple HTTP request/response exchanges to assemble the entire document.

## 25 HTTP and Caching

Versions 1.0 and 1.1 of the Hypertext Transfer Protocol (HTTP) are defined and described by the Internet Engineering Task Force (IETF) Request for Comment (RFC) 1945 by T. Berners-Lee, R. Fielding and H. Frystyk entitled "Hypertext Transfer Protocol -- HTTP/1.0", May, 1996 and in RFC 2068 by R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk and T. Berners-Lee  
30 entitled "Hypertext Transfer Protocol -- HTTP/1.1", January, 1997. Significant work has been done at the IETF as the HTTP protocol has evolved over the last 5 years from V1.0 to V1.1

recently adopted as the new standard. HTTP 1.1 has incorporated a number of enhancements to improve browser to web server and browser to web cache performance. The standard has historically and will continue to evolve at a necessarily slow pace to accommodate the millions of devices using the protocols to guarantee a smooth integration of the new capabilities.

5 As noted above, HTML pages and XML documents often reference one or more external entities which are separate physical storage objects and which must be fetched via the communications network in order to render or process the entire page or document. Because these external objects are frequently re-used by client applications, caching may be used to advantage. Historically, web browsers send multiple HTTP GET requests with the "If Modified-  
10 Since" extension to validate the current contents of the cache (local storage space holding recently fetched files for immediate recall). If the object in the local cache is still valid, the server responds with a "304" response directing the client application to use the local copy in the cache. If the content is not the same, the server returns a fresh copy of the data. The following text illustrates the verbose exchange required to retrieve a single external storage object using HTTP.

15 <Request>

GET http://www.ups.com/images/nav/global/ua\_upslogo\_globalnav.gif  
HTTP/1.1

20 Accept: \*/\*

Referer: http://www.ups.com/

Accept-Language: en-us

Accept-Encoding: gzip, deflate

If-Modified-Since: Thu, 08 Jun 1998 20:24:56 GMT; length=741

25 User-Agent: Mozilla/4.0 (compatible; MSIE 4.01; Windows 98)

Host: www.ups.com

Proxy-Connection: Keep-Alive

Pragma: No-Cache

Extension: Security/Remote-Passphrase

30 <Response>

HTTP/1.0 304 Use local copy

Server: Netscape-Enterprise/3.5.1C



Date: Tue, 06 Apr 1999 14:12:24 GMT

Link:

<http://www.ups.com/images/nav/global/ua\_upslogo\_globalnav.gif"  
PageServices>; rel="PageServices"

5 Connection: close

This lengthy exchange must be repeated for each object on the HTML page, even when those external files are already available on the local cache.

## 10 Description of the Best Mode

The present invention speeds the transmission and processing of data objects which incorporate other data objects by reference. More specifically, the present invention improves the performance of web servers, web caches and web browsers by repackaging existing HTML web page content for more efficient transport over the Internet using existing HTTP  
15 and TCP/IP facilities.

As contemplated by a first aspect of the present invention, means are employed on the server-side of the HTTP link for packaging coherency information for a given data object, such as an HTML web page or an XML document, together with coherency information describing the state of entities referenced in that page or document, so that the combined  
20 coherency information is sent to the client as a single object in a single HTTP request-response exchange. On the client side, the received object is interpreted to avoid the needless transmission of further HTTP requests when valid copies of referenced entities already exist in the local cache.

In accordance with another feature of the invention, means are employed on the server  
25 side for packaging the content of a given data object, such as an HTML web page or an XML document, together with content obtained from external resources referenced in that page or document, allowing the combined content to be transferred as a single object to the client. On the client side, the single received object is decomposed into its constituent parts which are may be rendered or processed without the need for further network transfers.

30 As contemplated by the invention, the client-side of an HTTP link may advantageously employ conventional web browsers, XML processors or other existing programs which are used in combination with an auxiliary program (e.g. a plug-in, helper, or proxy program) which acts as a transport layer between the HTTP interface and the application program, capturing and converting outbound HTTP requests from the application program into requests

for more concise objects, and for intercepting the incoming concise objects which are supplied by the server and translating them into the multiple data objects originally requested by the application program. Alternatively, client side applications, such as web browsers, may be modified to incorporate the functionality needed to directly request and handle the more  
5 concise combined objects contemplated by the invention.

In accordance with another aspect of the invention, means are employed at the server for processing data objects which incorporate other data objects by reference to create combined data objects for transmission to requesting clients. For example, a server-side program may be employed to process HTML web pages to create new combined data objects  
10 which include referenced data and which are requested by client side programs capable of processing those combined objects. Each such combined object provides the coherency data and content required to render an entire web page. Because these combined documents need be created only once, they can be built and stored asynchronous with the client requests and do not require significant server-side CPU resources.

In accordance with still further features of the invention, described in more detail below, the principles of the invention may be applied to improve the performance of Dynamic HTML (DHTML) web pages, including both Active Server Pages (ASP) and pages generated by Common Gateway Interface (CGI) programs. In addition, the invention may be used to implement a "delta cache" in which only the changed portions of modified programs or files  
15 need be sent from the server to the client, and for managing cache coherency by the use of unambiguous entity references.

These and other objects, features and advantages of the invention may be more fully understood through a consideration of the following detailed description.

## 25 Detailed Description

### Minimizing Cache Coherency Traffic

The principles of the present invention may be applied to advantage to eliminate unnecessary network request-response exchanges when valid copies of information included  
30 by reference in a given web page or XML document is already present in the local cache at the client. By way of example, if a web browser user clicks on a link to the URL "foo.htm", the browser issues an HTTP GET request for "foo.htm". Before the request is sent to the server, however, the client side program intercepts the request and instead issues a request to the web

server for a combined object named "oc\_foo.htm." The object named "oc\_foo.htm" may be produced by a server side program which combines foo.htm and other data objects which are to be included by reference in the rendered version of foo.htm. The combined object "oc\_foo.htm" contains coherency data for foo.htm and for the other objects referenced by  
5 foo.htm. As a consequence, the client program receives coherency information on all of the included objects in a single message exchange, eliminating the need for further exchanges to determine whether valid copies of the referenced objects are already present in the cache.

### Increasing Efficiency by Combining Content

10 According to a further feature of the invention, object packing is employed to efficiently transport data objects between the ICDS client and the ICDS server. Under conventional practice, each object required to render a web page must be individually requested by the browser and sent separately to the client by the web server; and each request is burdened by the overhead of the verbose HTTP protocol and latency of the Internet. As  
15 contemplated by the invention, server-side routines may be executed to pre-processes the HTML web page, combining the web page HTML with the content of data objects referenced by HTML tags, to form a combined object (e.g; a single file having its own URL) which includes a directory of offsets to the individual files objects contained within the combined object. The referenced objects included in the combined file may comprise image, applet or  
20 other HTML files specified in <OBJECT>, <IMG>, <APPLET>, <FRAME> or <IFRAME> tags.

There are a number of benefits in the combined packaging of content and coherency information as contemplated by the invention: the number of HTTP requests transmitted to the server is reduced, reducing the burden on the client, the server and on the network, and the  
25 cumulative delay caused by Internet latency is reduced since the number of Internet exchanges needed to display a given web page is reduced, regardless of whether or not that valid copies of that page and its referenced objects are present in the local cache.

If the browser already has some or many of the objects in the local cache, it would be a waste of time and bandwidth to transfer the entire combined object to get the required objects.  
30 In accordance with another feature of the invention, a feature of HTTP 1.1 which permits the transfer of designated file ranges may be used to advantage. The HTTP 1.1 protocol allows multiple designated ranges to be requested and transferred in a single request. As a consequence, when a combined coherency check indicates that the local cache can satisfy

some, but not all of the individual object requests, the objects which are still needed can be compiled into a compound request for objects not in the local cache.

It should also be noted that the combined objects created in accordance with this invention may be transmitted in a sequence of blocks sized to optimize network performance.

## 5 Optimization

Additional logic on the client-side and the server-side may be employed to optimize web page content delivery based on the current connection speed between the web browser and web server. For example, different classes of clients may issue requests for objects specifically optimized for those devices. For example, an object ID value may be transferred  
10 with the request which has one the values listed below:

- OCD\_ - ICDS-2 Meta Object (No optimization)
- OCD14\_ - Optimized for 14.4 connections
- OCD28\_ - Optimized for 28.8 connections
- 15 OCD56\_ - Optimized for 56K connections
- OCDT1\_ - Optimized for high speed T1 connections
- OCDTV\_ - Optimized for Web TV Delivery
- OCDPDA\_ - Optimized for PDA / Cell browsing

20 Web content may accordingly be pre-processed by the server-side routines which bind the constituent parts of web pages and referenced objects into combined objects that are optimized for particular client devices and which can be requested by these devices based on their connection speed and other criteria. The optimization of the transport process is transparent to the web page designers who may focus on the graphical content of the web site  
25 independent of the delivery speed over a wide range of connection speeds. The same technique can be used for adding additional capabilities including automatic language translation based on the clients requested language preference.

## Dynamic Web Content

There are a number of technologies collectively called server-side Dynamic HTML  
30 (DHTML) that provide facilities for dynamic rather than static responses to a browser request. Common applications for DHTML include customizing web pages to include current information (date/news/events) often driven by a database back-end. DHTML is also

commonly used to provide HTML page responses for search engines and e-commerce order processing.

In many applications only a small percentage of the DHTML generated by the server changes but the entire page must be retransmitted. (By definition, Dynamic HTML pages are not given modified dates and cannot be cached.) As contemplated by a further feature of the invention, a transparent mechanism may be employed to separate the static from the dynamic elements such that only the dynamic elements need be retrieved.

### Active Server Pages (ASP)

One popular type of Dynamic HTML, introduced by Microsoft, takes the form of Active Server Pages (ASP). ASP extends HTML to include tags that are redirected by the web server to an ASP process to convert the ASP requests into the desired HTML code. The browser only sees the resultant HTML code, not the ASP calls. As contemplated by the invention, Active Server Pages may be pre-processed to separate the active from the static elements on those pages. As a consequence, the server does not require an active process running on the web server to obtain the advantages of the invention. An example of the approach used is detailed below:

On the ASP server side:

1. A server program processes a static distribution of somefile1.ASP Create OC\_somefile1.asp including the last modified date of any other objects used to render the page.
2. Create OC\_Static\_somefile1.asp with #Include\_001 , #Include\_002 tags to replace the ASP calls.

Static Html

Static Html

<!-- #Include\_001-->

Static HTML

Static HTML

```
<!-- #Include_002-->
```

```
Static HTML
```

```
Static HTML
```

- 5           3.     Create OC\_Dynamic\_somefile1.asp file with identifying tags and the ASP calls

```
<!-- #Include_001 Start-->
```

```
ASP CODE
```

```
ASP CODE
```

10    ASP CODE

```
<!-- #Include_001 End-->
```

```
<!-- #Include_002 Start-->
```

```
ASP CODE
```

15    ASP CODE

```
ASP CODE
```

```
<!-- #Include_002 End-->
```

On the client side:

20

1.     Browser requests somefile1.asp (No modified date because it was uncacheable)
2.     OC client intercepts original request and instead requests OC\_somefile1.ASP  
       from the web server OC\_somefile1.asp contains coherency data for all  
25       objects including OC\_Static\_somefile1.asp )
3.     If OC\_Static\_somefile1.asp not in cache or not current, get  
       OC\_Static\_somefile1.asp
- 30     4.     Request OC\_Dynamic\_somefile1.asp from server (Never in local cache)
5.     OC client merges dynamic content returned from the server with static content  
       from local cache to produce the dynamic HTML page to the browser. (Also

marked as uncacheable to the browsers cache as it would have been in the original request.)

6. Note that the client side program responds to the original browser request for somefile1.asp and passes back the header information received from the server for OC\_Dynamic somefile1.asp

The use of the invention in connection with Active Server Pages (ASP) reduces the majority of the bytes necessary to refresh an ASP page because only the dynamic portions are actually transmitted between the server and the browser. There is no compromise of coherency. The redundant transfer of the static elements on ASP pages is eliminated, and no active process is required on the web server.

#### **CGI Based Dynamic HTML**

An earlier and still popular method for creating dynamic HTML pages is to use the Common Gateway Interface (CGI). CGI provides a standard way to execute programs on the web server to generate dynamic HTML code. CGI is very different from ASP because 1) the web server does not have access to the CGI application source code and 2) there are no standards for the actual code being run using the CGI interface. Nonetheless, the present invention can be used to advantage in connection with dynamic HTML implemented using CGI. To do so, an active process running on the web server is employed to intercept the HTML stream. This process parses the outgoing HTML stream and optimizes the information for transport over the Internet. By using the invention in connection with CGI processing, all of the objects referenced by the CGI generated HTML page can be checked for coherency with a single request and all of the data objects referenced by the CGI-generated HTML page can also be transferred in response to a single request. There is no compromise of coherency, and the need for multiple coherency checks and data - transfers of the static elements referenced on the CGI-generated pages is eliminated.



**Implementation of a "Delta Cach "**

Problems that can occur when more functionality is pushed down to the browser using client-side dynamic HTML. The advantage of client-side HTML code execution is that the client can do local processing and be more insulated from the web server/internet. The problem is that, as the programs transmitted for execution on the client get larger, even small changes to the program file require the entire file to be re-sent to the client.

As contemplated by the invention, this problem may be addressed by allowing the client to request the changes to a file instead of the entire file. On the server side, routines are provided which enable the server to identify those objects which may benefit significantly from using a delta cache, and "delta object" are only generated when a significant benefit (say, greater than a 70% saving) may be expected. On the client side, delta caching routines proceed as follows:

1. Browser requests foo.htm from web server
2. A client routine requests OC\_foo.htm from web server
3. The file oc\_foo.htm, contains coherency data for foo.htm and the objects referenced by foo.htm including the availability of delta objects.
4. A client routine determines that foo.htm or some other object referenced on the page and available in the client cache has changed
5. A client routine requests a combined object with delta changes
6. A client routine returns whole object with the delta changes to the browser
7. And the process is repeated for the other objects.

### A Global Object Store

As contemplated by still another feature of the invention, a global object store may be employed to solve one of the fundamental problems inherent in the manner in which information is currently distributed over the Internet. This problem stems from the ambiguous references which may be used by HTML to identify external entities which are to be included with the page. These ambiguous references require the browser to send additional requests to the server to determine if each file in the browser cache is the same as the file on the server. It should be noted that ambiguous references are a common problem that plagues web caches, wasting bandwidth by requiring continuous checking for newer version of each file on the web server.

Although the use of the coherency checking mechanism described above keeps redundant requests for coherency from propagating over the Internet, a further feature of the invention may be employed to reduce the processing overhead-itself. In accordance with this aspect of the invention, ambiguous file references are converted into non-ambiguous references by including the file coherency information as an integral part of the file name. Take for example a web page that references two graphics file banner1.gif and banner2.gif.

	banner1.gif	04/08/98	12:22:17
20	banner2.gif	05/15/99	09:07:13

To eliminate the ambiguous file reference problem, the file directory information shown above is automatically processed to rename the files as GOS\_S980804122217\_banner1.gif and GOS\_990515090713\_banner2.gif respectively, all HTML references to these files are changed to use the new names, the files are tagged as expiring 100 years from the processing date. As a consequence, if the web browser or web cache locates the file, it is by definition coherent and no further checks are required. Marking the file as expiring in 100 years makes use of a built in capability of HTTP that has long been underutilized.

### Accompanying Source Code

The accompanying Appendix contains source language listings of Java programs which implement some of the features of the invention as described above

and which have been developed using the Visual Café Java development environment supplied by Symantec Corporation of Cupertino, CA. In the Appendix, the Java class listings for the client side application are presented first, followed by the classes for the server side application, and concluding with common classes shared by both.

### **Conclusion**

It is to be understood that the methods which have been described are merely illustrative applications of the principles of the invention. Numerous modifications may be made without departing from the true spirit and scope of the invention.

```

/*
    Java Source Language Listings
    Copyright 1999 by Clinton Brenton Battersby
*/
/*
    A basic extension of the java.awt.Dialog class
*/

import java.awt.*;
import java.awt.image.ImageObserver;
import symantec.itools.awt.BorderPanel;

public class AboutClientDialog extends Dialog {

    public AboutClientDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setFont(new Font("Dialog", Font.PLAIN, 12));
        setSize(299,216);
        setVisible(false);
        try {
            borderPanel1.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanel1.setLayout(null);
        add(borderPanel1);
        borderPanel1.setBounds(2,0,298,188);
        label4.setText("Patents pending");
        borderPanel1.add(label4);
        label4.setBounds(20,65,156,24);
        label3.setText("All Rights Reserved");
        borderPanel1.add(label3);
        label3.setBounds(20,45,156,24);
        label2.setText("Copyright (C) 1999 MTI");
        borderPanel1.add(label2);
        label2.setBounds(21,30,156,20);
        label5.setText("Customer:");
        borderPanel1.add(label5);
        label5.setBounds(20,103,78,24);
        label6.setText("Version: 2");
        borderPanel1.add(label6);
        label6.setBounds(20,125,77,19);
        label7.setText("Build: ");
        borderPanel1.add(label7);
        label7.setBounds(20,145,48,18);
        borderPanel1.add(textField1);
        textField1.setBounds(96,105,173,20);
        buildlabel.setText("0");
        borderPanel1.add(buildlabel);
        buildlabel.setBounds(71,145,88,15);
        button1.setLabel("WWWhoosh! Client Accelerator");
        borderPanel1.add(button1);
        button1.setBounds(13,8,185,19);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        okButton.setBounds(227,189,64,20);
        setTitle("About WWWhoosh Client");
        //}}
    }
}

```

```

        buildLabel.setText( new Integer(ClientBuild.NUMBER).toString() );
    }
    //((REGISTER_LISTENERS
    SymWindow aSymWindow = new SymWindow();
    this.addWindowListener(aSymWindow);
    SymAction lSymAction = new SymAction();
    okButton.addActionListener(lSymAction);
    button1.addActionListener(lSymAction);
    //))

}

public AboutClientDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

public void paint(Graphics g)
{
    Image logo = getToolkit().getImage("Javalogo.jpg");
    g.drawImage(logo, 15, 35, 65, 65, (ImageObserver) this);
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }

    super.setVisible(b);
}

//((DECLARE_CONTROLS
symantec.itools.awt.BorderPanel borderPanel1 = new symantec.itools.awt.BorderPanel();
java.awt.Label label4 = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
java.awt.Label label2 = new java.awt.Label();
java.awt.Label label5 = new java.awt.Label();

```

```

java.awt.Label label6 = new java.awt.Label();
java.awt.Label label7 = new java.awt.Label();
java.awt.TextField textField1 = new java.awt.TextField();
java.awt.Label buildlabel = new java.awt.Label();
java.awt.Button button1 = new java.awt.Button();
java.awt.Button okButton = new java.awt.Button();
//))

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
        else if (object == button1)
            button1_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == AboutClientDialog.this)
            AboutClientDialog_WindowClosing(event);
    }
}

void AboutClientDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    AboutClientDialog_WindowClosing_Interaction1(event);
}

void AboutClientDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

static java.util.ResourceBundle aboutClientDialogBundle =
java.util.ResourceBundle.getBundle("AboutClientDialogBundle");

```

```
void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    Runtime rt = Runtime.getRuntime();
    try {
        rt.exec("C:\\Program Files\\Internet Explorer\\Iexplore.exe
http://www.wwwhoosh.com");
    }
    catch (java.io.IOException ice)
    {}
}
```



```
public class AboutClientDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //((Start Automatic Code Generation Block
        {"borderPanel1_label", ""},
        {"okButton_label", "OK"},
        {"label4_text", "Patents pending"},
        {"label3_text", "All Rights Reserved"},
        {"label2_text", "Copyright (C) 1999 MTI"},
        {"label1_text", "WWWhoosh Proxy"},
        {"label5_text", "Customer:"},
        {"label6_text", "Version: 001"},
        {"label7_text", "Build: "},
        {"textField1_text", ""},
        {"buildlabel_text", "0"},
        {"AboutClientDialog_title", "About WWWhoosh Client"},
        //))End Automatic Code Generation Block
    };
}
```

```

import java.awt.*;
import java.awt.event.*;
import COM.reitshamer.util.Debug;
import symantec.itools.awt.BorderPanel;

public class ClearDialog extends Dialog
{
    public ClearDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        //Keep a local reference to the invoking frame
        frame = parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(214,133);
        setVisible(false);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(9,7,200,95);
        label1.setText("Clear Log to Continue?");
        label1.setAlignment(java.awt.Label.CENTER);
        borderPanell.add(label1);
        label1.setBounds(-1,27,180,23);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        okButton.setBounds(136,104,64,20);
        setTitle("Screen Log Full");
        //}}

        //{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        okButton.addActionListener(lSymAction);
        //}}
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;

        // Adjust components according to the insets
        setSize(getInsets().left + getInsets().right + d.width, getInsets().top + getInsets().bottom
+ d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(getInsets().left, getInsets().top);
        }
    }
}

```

```

        components[i].setLocation(p);
    }
    fComponentsAdjusted = true;
}

public ClearDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

/**
 * Shows or hides the component depending on the boolean flag b.
 * @param b if true, show the component; otherwise, hide the component.
 * @see java.awt.Component#isVisible
 */
public void setVisible(boolean b)
{
    if(b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width)/ 2,
            bounds.y + (bounds.height - abounds.height)/2);
        Toolkit.getDefaultToolkit().beep();
    }
    super.setVisible(b);
}

// Used for addNotify check.
boolean fComponentsAdjusted = false;
// Invoking frame
Frame frame = null;

//{{{DECLARE_CONTROLS
symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
java.awt.Label labell = new java.awt.Label();
java.awt.Button okButton = new java.awt.Button();
//}}}

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        Debug.textout.setText("");

        this.dispose();
    } catch (Exception e) {
    }
}

```

```
class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == ClearDialog.this)
            ClearDialog_WindowClosing(event);
    }
}

void ClearDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    ClearDialog_WindowClosing_Interaction1(event);
}

void ClearDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

static java.util.ResourceBundle clearDialogBundle =
java.util.ResourceBundle.getBundle("ClearDialogBundle");
}
```

```
public class ClearDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //{{Start Automatic Code Generation Block
        {"okButton_label", "OK"},
        {"label1_text", "Clear Log to Continue?"},
        {"ClearDialog_title", "Screen Log Full"},
        //}}End Automatic Code Generation Block
    };
}
```

```
public class ClientBuild {public static int NUMBER=143 ; }
```

```
public class ClearDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //{{Start Automatic Code Generation Block
        {"okButton_label", "OK"},
        {"label1_text", "Clear Log to Continue?"},
        {"ClearDialog_title", "Screen Log Full"},
        //}}End Automatic Code Generation Block
    };
}
```



```

import java.awt.*;

public class CommandHelpDialog extends Dialog {
    public CommandHelpDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //((INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.white);
        setFont(new Font("Dialog", Font.PLAIN, 14));
        setSize(446,203);
        setVisible(false);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setBackground(java.awt.Color.lightGray);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        okButton.setBounds(200,164,60,20);
        label1.setText("Passthrough: forward all requests unmodified, no check cache");
        add(label1);
        label1.setFont(new Font("Dialog", Font.PLAIN, 12));
        label1.setBounds(12,42,420,36);
        label2.setText("Debug trace: turns HTTP logging on/off");
        add(label2);
        label2.setFont(new Font("Dialog", Font.PLAIN, 12));
        label2.setBounds(12,72,420,36);
        label3.setText("Loopback: responds to ALL requests with \"304 Not Modified\"");
        add(label3);
        label3.setFont(new Font("Dialog", Font.PLAIN, 12));
        label3.setBounds(12,12,420,36);
        setTitle("WWWhoosh commands");
        //))

        //((REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        okButton.addActionListener(lSymAction);
        //))
    }

    public CommandHelpDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        // Only do this once.
        if (fComponentsAdjusted)
            return;

        // Adjust components according to the insets
    }

```

```

Insets insets = getInsets();
setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
Component components[] = getComponents();
for (int i = 0; i < components.length; i++)
{
    Point p = components[i].getLocation();
    p.translate(insets.left, insets.top);
    components[i].setLocation(p);
}

// Used for addNotify check.
fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }

    super.setVisible(b);
}

//{{{DECLARE_CONTROLS
java.awt.Button okButton = new java.awt.Button();
java.awt.Label label1 = new java.awt.Label();
java.awt.Label label2 = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
//}}}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {

```

```
        Object object = event.getSource();
        if (object == CommandHelpDialog.this)
            CommandHelpDialog_WindowClosing(event);
    }

    void CommandHelpDialog_WindowClosing(java.awt.event.WindowEvent event)
    {
        // to do: code goes here.

        CommandHelpDialog_WindowClosing_Interaction1(event);
    }

    void CommandHelpDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
    {
        try {
            this.dispose();
        } catch (Exception e) {
        }
    }

    static java.util.ResourceBundle commandHelpDialogBundle =
    java.util.ResourceBundle.getBundle("CommandHelpDialogBundle");
}
```

```
public class CommandHelpDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //{{Start Automatic Code Generation Block
        {"okButton_label", "OK"},
        {"label1_text", "Passthrough: forward all requests unmodified, no check cache"},
        {"label2_text", "Debug trace: turns HTTP logging on/off"},
        {"label3_text", "Loopback: responds to ALL requests with \"304 Not Modified\""},
        {"CommandHelpDialog_title", "WWWhoosh commands"},
        //}}End Automatic Code Generation Block
    };
}
```

```

import java.awt.*;
import symantec.itools.awt.BorderPanel;
import COM.reitshamer.util.*;

public class ConnectionDialog extends Dialog {
    private ProxyGUI _proxy=null;
    public ConnectionDialog(ProxyGUI parent, boolean modal)
    {
        super(parent, modal);
        _proxy=parent;
        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setFont(new Font("Dialog", Font.PLAIN, 14));
        setSize(373,299);
        setVisible(false);
        button1.setLabel("OK");
        add(button1);
        button1.setFont(new Font("Dialog", Font.PLAIN, 12));
        button1.setBounds(148,271,64,20);
        button2.setLabel("Cancel");
        add(button2);
        button2.setFont(new Font("Dialog", Font.PLAIN, 12));
        button2.setBounds(215,271,64,20);
        button3.setLabel("Apply");
        add(button3);
        button3.setFont(new Font("Dialog", Font.PLAIN, 12));
        button3.setBounds(279,271,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setLabel("  Internet Connection  ");
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setFont(new Font("Dialog", Font.PLAIN, 12));
        borderPanell.setBounds(4,11,342,156);
        proxycheck.setLabel("Connect using a proxy server ");
        borderPanell.add(proxycheck);
        proxycheck.setFont(new Font("Dialog", Font.PLAIN, 12));
        proxycheck.setBounds(11,11,228,24);
        addressText.setText("127.0.0.1");
        borderPanell.add(addressText);
        addressText.setBounds(102,45,112,20);
        borderPanell.add(portText);
        portText.setBounds(264,45,52,20);
        label2.setText(" Address:");
        borderPanell.add(label2);
        label2.setFont(new Font("Dialog", Font.PLAIN, 12));
        label2.setBounds(47,41,56,24);
        label3.setText("Port:");
        borderPanell.add(label3);
        label3.setBounds(227,45,36,20);
        try {

```

```

        borderPanel2.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
    }
    catch(java.beans.PropertyVetoException e) { }
    try {
        borderPanel2.setLabel("  Browser Connection  ");
    }
    catch(java.beans.PropertyVetoException e) { }
    try {
        borderPanel2.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
    }
    catch(java.beans.PropertyVetoException e) { }
    borderPanel2.setLayout(null);
    add(borderPanel2);
    borderPanel2.setFont(new Font("Dialog", Font.PLAIN, 12));
    borderPanel2.setBounds(10,174,338,89);
    label4.setText("Listen for connections on port:");
    borderPanel2.add(label4);
    label4.setBounds(8,4,172,28);
    listenportText.setText("123");
    borderPanel2.add(listenportText);
    listenportText.setBounds(210,9,52,20);
    setTitle("WWWhoosh Connection Options");
    //}}

    //{{REGISTER_LISTENERS
    SymWindow aSymWindow = new SymWindow();
    this.addWindowListener(aSymWindow);
    SymAction lSymAction = new SymAction();
    button1.addActionListener(lSymAction);
    SymComponent aSymComponent = new SymComponent();
    this.addComponentListener(aSymComponent);
    button2.addActionListener(lSymAction);
    button3.addActionListener(lSymAction);
    SymItem lSymItem = new SymItem();
    proxycheck.addItemListener(lSymItem);
    //}}
}

public ConnectionDialog(ProxyGUI parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

```

```

    }

    public void setVisible(boolean b)
    {
        if (b)
        {
            Rectangle bounds = getParent().getBounds();
            Rectangle abounds = getBounds();

            setLocation(bounds.x + (bounds.width - abounds.width) / 2,
                bounds.y + (bounds.height - abounds.height) / 2);
        }

        super.setVisible(b);
    }

    /**(DECLARE_CONTROLS
    java.awt.Button button1 = new java.awt.Button();
    java.awt.Button button2 = new java.awt.Button();
    java.awt.Button button3 = new java.awt.Button();
    symantec.itools.awt.BorderPanel borderPanel1 = new symantec.itools.awt.BorderPanel();
    java.awt.Checkbox proxycheck = new java.awt.Checkbox();
    java.awt.TextField addressText = new java.awt.TextField();
    java.awt.TextField portText = new java.awt.TextField();
    java.awt.Label label2 = new java.awt.Label();
    java.awt.Label label3 = new java.awt.Label();
    symantec.itools.awt.BorderPanel borderPanel2 = new symantec.itools.awt.BorderPanel();
    java.awt.Label label4 = new java.awt.Label();
    java.awt.TextField listenportText = new java.awt.TextField();
    /**)

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;

    class SymAction implements java.awt.event.ActionListener
    {
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
            Object object = event.getSource();
            if (object == button1)
                button1_ActionPerformed(event);
            else if (object == button2)
                button2_ActionPerformed(event);
            else if (object == button3)
                button3_ActionPerformed(event);
        }
    }

    class SymWindow extends java.awt.event.WindowAdapter
    {
        public void windowClosed(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == ConnectionDialog.this)
                ConnectionDialog_WindowClosed(event);
        }

        public void windowOpened(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == ConnectionDialog.this)
                ConnectionDialog_WindowOpened(event);
        }

        public void windowClosing(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();

```



```

        if (object == ConnectionDialog.this)
            ConnectionDialog_WindowClosing(event);
    }

void ConnectionDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    ConnectionDialog_WindowClosing_Interaction1(event);
}

void ConnectionDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    Debug.useProxy=proxycheck.getState();
    if (proxycheck.getState() && (portText.getText() != "")) {
        {
            Debug.outAddress= addressText.getText();
            Debug.outProxy=(new Integer(portText.getText())).intValue();
        }

        if (listenportText.getText() != "")
        {
            int newport = ( new Integer(listenportText.getText()) ).intValue();
            if (Debug.listenPort != newport)
            {
                Debug.listenPort = newport;
                try {_proxy.init();}
                catch (Exception e)
                {}
            }
        }
        try {
            this.dispose();
        } catch (Exception e) {
        }
    }
}

class SymComponent extends java.awt.event.ComponentAdapter
{
    public void componentShown(java.awt.event.ComponentEvent event)
    {
        Object object = event.getSource();
        if (object == ConnectionDialog.this)
            ConnectionDialog_ComponentShown(event);
    }
}

void ConnectionDialog_ComponentShown(java.awt.event.ComponentEvent event)
{
}

void ConnectionDialog_WindowOpened(java.awt.event.WindowEvent event)
{
    listenportText.setText(new Integer(Debug.listenPort).toString() );
    proxycheck.setState(Debug.useProxy);
}

```

```

        addressText.setText(Debug.outAddress );
        portText.setText(new Integer(Debug.outProxy).toString() );

        if (!Debug.useProxy)
        {
            portText.disable();
            addressText.disable();
        }
    }

    void button2_ActionPerformed(java.awt.event.ActionEvent event)
    {
        try {
            this.dispose();
        } catch (Exception e) {
        }
    }

    void button3_ActionPerformed(java.awt.event.ActionEvent event)
    {
        Debug.useProxy=proxycheck.getState();

        if (proxycheck.getState() && (portText.getText() != ""))
        {
            Debug.outAddress= addressText.getText();
            Debug.outProxy=(new Integer(portText.getText())).intValue();
        }

        if (listenportText.getText() != "")
        {
            int newport = ( new Integer(listenportText.getText()) ).intValue();
            if (Debug.listenPort != newport)
            {
                Debug.listenPort = newport;
                try {_proxy.init();}
                catch (Exception e)
                {}
            }
        }
    }

    void ConnectionDialog_WindowClosed(java.awt.event.WindowEvent event)
    {
        // to do: code goes here.
    }

    class SymItem implements java.awt.event.ItemListener
    {
        public void itemStateChanged(java.awt.event.ItemEvent event)
        {
            Object object = event.getSource();
            if (object == proxycheck)
                proxycheck_ItemStateChanged(event);
        }
    }

    void proxycheck_ItemStateChanged(java.awt.event.ItemEvent event)
    {
        if (proxycheck.getState() )
        {
            portText.enable();
            addressText.enable();
        }
        else
    }

```

```
        {portText.disable();  
          addressText.disable();}  
    }  
    static      java.util.ResourceBundle      connectionDialogBundle      =  
    java.util.ResourceBundle.getBundle("ConnectionDialogBundle");  
}
```

```
public class ConnectionDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //{{Start Automatic Code Generation Block
        {"label1_text", " Connection Options "},
        {"button1_label", "OK"},
        {"button2_label", "Cancel"},
        {"button3_label", "Apply"},
        {"borderPanell_label", ""},
        {"proxycheck_label", "Connect using a proxy server "},
        {"addressText_text", ""},
        {"portText_text", ""},
        {"label2_text", " Address:"},
        {"label3_text", "Port:"},
        {"listenportText_text", ""},
        {"label4_text", "Listen for connections on port:"},
        {"ConnectionDialog_title", "WWWwhoosh Connection Options"},
        //}}End Automatic Code Generation Block
    };
}
```

```

import java.awt.*;
import symantec.itools.awt.BorderPanel;
import COM.reitshamer.util.*;
import com.symantec.itools.swing.JButtonGroupPanel;

public class FeatureDialog extends Dialog {
    public FeatureDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //((INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setFont(new Font("Dialog", Font.PLAIN, 14));
        setSize(325,144);
        setVisible(false);
        label1.setText(" Functions ");
        add(label1);
        label1.setFont(new Font("Dialog", Font.PLAIN, 12));
        label1.setBounds(24,12,92,28);
        button1.setLabel("OK");
        add(button1);
        button1.setFont(new Font("Dialog", Font.PLAIN, 12));
        button1.setBounds(113,114,64,20);
        button2.setLabel("Cancel");
        add(button2);
        button2.setFont(new Font("Dialog", Font.PLAIN, 12));
        button2.setBounds(178,114,64,20);
        button3.setLabel("Apply");
        add(button3);
        button3.setFont(new Font("Dialog", Font.PLAIN, 12));
        button3.setBounds(242,114,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(12,13,306,98);
        statsthreadCheckbox.setLabel("HTTP 1.1 Browser Connections");
        borderPanell.add(statsthreadCheckbox);
        statsthreadCheckbox.setFont(new Font("Dialog", Font.PLAIN, 12));
        statsthreadCheckbox.setBounds(14,21,221,18);
        forceGetCheckbox.setState(true);
        forceGetCheckbox.setLabel("Force GET of HTML base");
        borderPanell.add(forceGetCheckbox);
        forceGetCheckbox.setFont(new Font("Dialog", Font.PLAIN, 12));
        forceGetCheckbox.setBounds(14,45,221,18);
        setTitle("WWWhoosh Optional Functions");
        //))

        //((REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        button1.addActionListener(lSymAction);
        SymComponent aSymComponent = new SymComponent();
        this.addComponentListener(aSymComponent);
        button2.addActionListener(lSymAction);
        button3.addActionListener(lSymAction);
    }
}

```

```

    ///}
}

public FeatureDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    //force noresize
    setResizable(false);

    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }

    super.setVisible(b);
}

//{(DECLARE CONTROLS
java.awt.Label label1 = new java.awt.Label();
java.awt.Button button1 = new java.awt.Button();
java.awt.Button button2 = new java.awt.Button();
java.awt.Button button3 = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
java.awt.Checkbox statsthreadCheckbox = new java.awt.Checkbox();
java.awt.Checkbox forceGetCheckbox = new java.awt.Checkbox();
///}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener

```

```

(
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == button1)
            button1_ActionPerformed(event);
        else if (object == button2)
            button2_ActionPerformed(event);
        else if (object == button3)
            button3_ActionPerformed(event);
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosed(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == FeatureDialog.this)
            FeatureDialog_WindowClosed(event);
    }

    public void windowOpened(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == FeatureDialog.this)
            FeatureDialog_WindowOpened(event);
    }

    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == FeatureDialog.this)
            FeatureDialog_WindowClosing(event);
    }
}

void FeatureDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    FeatureDialog_WindowClosing_Interaction1(event);
}

void FeatureDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    Debug.connKeepAlive = statsthreadCheckbox.getState();
    Debug.forceHTMLbaseGET=forceGetCheckbox.getState();

    try {
        this.dispose();
    } catch (Exception e) {
    }
}

```

```

    }

    class SymComponent extends java.awt.event.ComponentAdapter
    {
        public void componentShown(java.awt.event.ComponentEvent event)
        {
            Object object = event.getSource();
            if (object == FeatureDialog.this)
                FeatureDialog_ComponentShown(event);
        }
    }

    void FeatureDialog_ComponentShown(java.awt.event.ComponentEvent event)
    {
    }

    void FeatureDialog_WindowOpened(java.awt.event.WindowEvent event)
    {
        statsthreadCheckbox.setState( Debug.connKeepAlive );
        forceGetCheckbox.setState( Debug.forceHTMLbaseGET );
    }

    void button2_ActionPerformed(java.awt.event.ActionEvent event)
    {
        try {
            this.dispose();
        } catch (Exception e) {
        }
    }

    void button3_ActionPerformed(java.awt.event.ActionEvent event)
    {
        Debug.connKeepAlive = statsthreadCheckbox.getState();
        Debug.forceHTMLbaseGET=forceGetCheckbox.getState();
    }

    void FeatureDialog_WindowClosed(java.awt.event.WindowEvent event)
    {
        // to do: code goes here.
    }

    class SymItem implements java.awt.event.ItemListener
    {
        public void itemStateChanged(java.awt.event.ItemEvent event)
        {
        }
    }

    static java.util.ResourceBundle featuredDialogBundle =
    java.util.ResourceBundle.getBundle("FeatureDialogBundle");
}

```



```
public class FeatureDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //{{Start Automatic Code Generation Block
        {"label1_text", " Functions "},
        {"borderPanel1_label", ""},
        {"checkbox1_label", "Statistics Update Thread"},
        {"button1_label", "OK"},
        {"button2_label", "Cancel"},
        {"button3_label", "Apply"},
        {"FeatureDialog_title", "WWWhoosh Log Levels"},
        //}}End Automatic Code Generation Block
    };
}
```

```

/**
 * A basic extension of the java.awt.Frame class
 */

import java.awt.*;
import COM.reitshamer.util.Debug;
import java.util.Date;

public class LoggerFrame extends Frame
{
    public LoggerFrame()
    {
        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(526,442);
        setVisible(false);
        button1.setLabel("Close");
        add(button1);
        button1.setBounds(451,411,64,20);
        add(textAreal);
        textAreal.setBounds(11,9,507,392);
        button2.setLabel("Clear");
        add(button2);
        button2.setBounds(387,411,64,20);
        button3.setLabel("Logfile");
        add(button3);
        button3.setBounds(322,411,64,20);
        setTitle("WWWhoosh Trace Logger ");
        setResizable(false);
        //}}}

        //{{{INIT_MENUS
        //}}}

        //{{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        button1.addActionListener(lSymAction);
        button3.addActionListener(lSymAction);
        button2.addActionListener(lSymAction);
        //}}}
    }

    public LoggerFrame(String title)
    {
        this();
        setTitle(title);
    }

    /**
     * Shows or hides the component depending on the boolean flag b.
     * @param b if true, show the component; otherwise, hide the component.
     * @see java.awt.Component#isVisible
     */
    public void setVisible(boolean b)
    {
        textAreal = Debug.textout;

        if(b)
        {
            setLocation(300, 300);
        }
    }
}

```

```

        super.setVisible(b);
    }

    static public void main(String args[])
    {
        (new LoggerFrame()).setVisible(true);
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;

        // Adjust components according to the insets
        Insets insets = getInsets();
        setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(insets.left, insets.top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;

    {{{DECLARE_CONTROLS
    java.awt.Button button1 = new java.awt.Button();
    java.awt.TextArea textArea1 = new java.awt.TextArea();
    java.awt.Button button2 = new java.awt.Button();
    java.awt.Button button3 = new java.awt.Button();
    }}}

    {{{DECLARE_MENUS
    }}}

    class SymWindow extends java.awt.event.WindowAdapter
    {
        public void windowClosing(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == LoggerFrame.this)
                LoggerFrame_WindowClosing(event);
        }
    }

    void LoggerFrame_WindowClosing(java.awt.event.WindowEvent event)
    {
        setVisible(false);    // hide the Frame
    }

    class SymAction implements java.awt.event.ActionListener
    {
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
            Object object = event.getSource();
            if (object == button1)
                button1_ActionPerformed(event);
        }
    }

```

```
        else if (object == button3)
            button3_ActionPerformed(event);
        else if (object == button2)
            button2_ActionPerformed(event);
    }
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    button1_ActionPerformed_Interaction1(event);
}

void button1_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // LoggerFrame Hide the LoggerFrame
        this.setVisible(false);
    } catch (Exception e) {
    }
}

void button3_ActionPerformed(java.awt.event.ActionEvent event)
{
    if (Debug.fileOut() )
    {
        Debug.forceprint("\r\n" + "File output stopped at " + new Date().toGMTString() + "\r\n");
    };
    Debug.setLogOff();
}
else
{
    Debug.setLogOn();
    Debug.forceprint("\r\n" + "File output started at " + new Date().toGMTString() + "\r\n");
};
}

void button2_ActionPerformed(java.awt.event.ActionEvent event)
{
    Debug.textout.setText(" ");
}
}
```

```

import java.awt.*;
import symantec.itools.awt.BorderPanel;
import COM.reitshamer.util.*;

public class LogLevelDialog extends Dialog {

    public LogLevelDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setFont(new Font("Dialog", Font.PLAIN, 14));
        setSize(350,222);
        setVisible(false);
        try {
            borderPanel1.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel1.setIPadBottom(2);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel1.setLabel("Logging Options");
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel1.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanel1.setLayout(null);
        add(borderPanel1);
        borderPanel1.setFont(new Font("Dialog", Font.PLAIN, 12));
        borderPanel1.setBounds(6,12,338,177);
        timingcheck.setLabel("Timing and Statistical Info");
        borderPanel1.add(timingcheck);
        timingcheck.setFont(new Font("Dialog", Font.PLAIN, 12));
        timingcheck.setBounds(5,108,308,20);
        headerscheck.setLabel("Print HTTP Headers");
        borderPanel1.add(headerscheck);
        headerscheck.setFont(new Font("Dialog", Font.PLAIN, 12));
        headerscheck.setBounds(5,68,318,20);
        logiccheck.setLabel("Trace Cache Logic Flow");
        borderPanel1.add(logiccheck);
        logiccheck.setFont(new Font("Dialog", Font.PLAIN, 12));
        logiccheck.setBounds(5,88,272,20);
        getscheck.setLabel("HTTP Request and Response Status Lines");
        borderPanel1.add(getscheck);
        getscheck.setFont(new Font("Dialog", Font.PLAIN, 12));
        getscheck.setBounds(5,28,312,20);
        fullpathCheckbox.setLabel("Display Full Pathnames");
        borderPanel1.add(fullpathCheckbox);
        fullpathCheckbox.setFont(new Font("Dialog", Font.PLAIN, 12));
        fullpathCheckbox.setBounds(5,48,186,20);
        checkbox1.setLabel("Show Exceptions");
        borderPanel1.add(checkbox1);
        checkbox1.setFont(new Font("Dialog", Font.PLAIN, 12));
        checkbox1.setBounds(5,8,316,20);
        button1.setLabel("OK");
        add(button1);
        button1.setFont(new Font("Dialog", Font.PLAIN, 12));
    }
}

```

```

        button1.setBounds(139,192,64,20);
        button2.setLabel("Cancel");
        add(button2);
        button2.setFont(new Font("Dialog", Font.PLAIN, 12));
        button2.setBounds(205,192,64,20);
        button3.setLabel("Apply");
        add(button3);
        button3.setFont(new Font("Dialog", Font.PLAIN, 12));
        button3.setBounds(269,192,64,20);
        setTitle("WWWhoosh Log Levels");
    }

    //((REGISTER_LISTENERS
    SymWindow aSymWindow = new SymWindow();
    this.addWindowListener(aSymWindow);
    SymAction lSymAction = new SymAction();
    button1.addActionListener(lSymAction);
    SymComponent aSymComponent = new SymComponent();
    this.addComponentListener(aSymComponent);
    button2.addActionListener(lSymAction);
    button3.addActionListener(lSymAction);
    SymItem lSymItem = new SymItem();
    fullPathCheckbox.addItemListener(lSymItem);
    //))

}

public LogLevelDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width)/ 2,
            bounds.y + (bounds.height - abounds.height)/2);
    }
}

```

```

        super.setVisible(b);
    }

    /**(DECLARE_CONTROLS
    symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
    java.awt.Checkbox timingcheck = new java.awt.Checkbox();
    java.awt.Checkbox headerscheck = new java.awt.Checkbox();
    java.awt.Checkbox logiccheck = new java.awt.Checkbox();
    java.awt.Checkbox getscheck = new java.awt.Checkbox();
    java.awt.Checkbox fullpathCheckbox = new java.awt.Checkbox();
    java.awt.Checkbox checkbox1 = new java.awt.Checkbox();
    java.awt.Button button1 = new java.awt.Button();
    java.awt.Button button2 = new java.awt.Button();
    java.awt.Button button3 = new java.awt.Button();
    /**)

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;

    class SymAction implements java.awt.event.ActionListener
    {
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
            Object object = event.getSource();
            if (object == button1)
                button1_ActionPerformed(event);
            else if (object == button2)
                button2_ActionPerformed(event);
            else if (object == button3)
                button3_ActionPerformed(event);
        }
    }

    class SymWindow extends java.awt.event.WindowAdapter
    {
        public void windowClosed(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == LogLevelDialog.this)
                LogLevelDialog_WindowClosed(event);
        }

        public void windowOpened(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == LogLevelDialog.this)
                LogLevelDialog_WindowOpened(event);
        }

        public void windowClosing(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == LogLevelDialog.this)
                LogLevelDialog_WindowClosing(event);
        }
    }

    void LogLevelDialog_WindowClosing(java.awt.event.WindowEvent event)
    {
        // to do: code goes here.

        LogLevelDialog_WindowClosing_Interaction1(event);
    }

    void LogLevelDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
    {

```

```

        try {
            this.dispose();
        } catch (Exception e) {
        }
    }

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    Debug.LogGETs=getscheck.getState();
    Debug.LogCache=logiccheck.getState();
    Debug.LogHeaders=headerscheck.getState();
    Debug.FULLPATHS=fullpathCheckbox.getState();

    Debug.LogTiming=timingcheck.getState();
    try {
        this.dispose();
    } catch (Exception e) {
    }

}

class SymComponent extends java.awt.event.ComponentAdapter
{
    public void componentShown(java.awt.event.ComponentEvent event)
    {
        Object object = event.getSource();
        if (object == LogLevelDialog.this)
            LogLevelDialog_ComponentShown(event);
    }
}

void LogLevelDialog_ComponentShown(java.awt.event.ComponentEvent event)
{
}

void LogLevelDialog_WindowOpened(java.awt.event.WindowEvent event)
{
    getscheck.setState( Debug.LogGETs);
    logiccheck.setState( Debug.LogCache);
    headerscheck.setState( Debug.LogHeaders);
    timingcheck.setState( Debug.LogTiming);
    fullpathCheckbox.setState(Debug.COMPRESS);
}

void button2_ActionPerformed(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

void button3_ActionPerformed(java.awt.event.ActionEvent event)
{
    Debug.LogGETs=getscheck.getState();
    Debug.LogCache=logiccheck.getState();
    Debug.LogHeaders=headerscheck.getState();
    Debug.COMPRESS=fullpathCheckbox.getState();
    //Debug.LogHeaders=timingcheck.getState();
}

void LogLevelDialog_WindowClosed(java.awt.event.WindowEvent event)
{
}

```



```
// to do: code goes here.
}

class SymItem implements java.awt.event.ItemListener
{
    public void itemStateChanged(java.awt.event.ItemEvent event)
    {
        Object object = event.getSource();
        if (object == fullpathCheckbox)
            fullpathCheckbox_ItemStateChanged(event);
    }
}

void fullpathCheckbox_ItemStateChanged(java.awt.event.ItemEvent event)
{
    // to do: code goes here.
    Debug.FULLPATHS = fullpathCheckbox.getState();
}

static java.util.ResourceBundle logLevelDialogBundle =
java.util.ResourceBundle.getBundle("LogLevelDialogBundle");
}
```

```
public class LogLevelDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //((Start Automatic Code Generation Block
        {"label1_text", " Log Options"},
        {"borderPanell_label", ""},
        {"timingcheck_label", "Timing, statistics info"},
        {"headerscheck_label", "Print HTTP headers"},
        {"logiccheck_label", "Trace cache logic flow"},
        {"getscheck_label", "GET requests, server Responses, 304 responses"},
        {"fullpathCheckbox_label", "Display full pathnames"},
        {"checkboxx1_label", "Show Exceptions"},
        {"button1_label", "OK"},
        {"button2_label", "Cancel"},
        {"button3_label", "Apply"},
        {"LogLevelDialog_title", "WWWhoosh Log Levels"},
        //))End Automatic Code Generation Block
    };
}
```

```

import java.net.*;
import java.io.*;
import COM.reitshamer.http.OCDResponse;
import COM.reitshamer.error.*;
import COM.reitshamer.util.*;
import COM.reitshamer.dc.*;
/***** OCDReaderThread *****/
//
// @author Mark Palmer
// @version 0.1 - initial beta release, 1-JUL-1999
//
// Modifications:
//
//
/***** OCDReaderThread *****/

/*
 * @input _sock      Socket that the incoming request is coming in on
 * @input ocdR - the OCD Response to be processed via this socket
 */
public class OCDReaderThread extends Thread
{
    private Socket sock;
    private OCDResponse _ocdr;

    public OCDReaderThread(Socket _sock, OCDResponse ocdR)
    {
        super();
        sock = _sock;
        _ocdr = ocdR;
        //{{INIT_CONTROLS
        //}}
    }

    public void run()
    {
        try
        {
            _ocdr.read(sock);
        }
        catch(HmrException e)
        {
            Debug.forceprint("OCD reader thread exception: \n " + e.getMessage() );
            try
            {
                sock.close();
            }
            catch(IOException ioe) {}
        }
    }
    //{{DECLARE_CONTROLS
    //}}
}

```

```

import WWWhooshCommon.*;
import java.awt.*;
import java.util.Hashtable;
import COM.reitshamer.util.*;
import COM.reitshamer.net.*;
import COM.reitshamer.error.*;
import COM.reitshamer.http.*;
import COM.reitshamer.proxy.*;

import java.util.Date;
import java.io.*;
import symantec.itools.awt.BorderPanel;
import symantec.itools.awt.ImageButton;

/***** PROXY GUI *****/
//
// @author Mark Palmer
// @version 2.0 - GUI revision
//
// Modifications:
//
// Build      Date      Description
//
// 1.995      02-Aug-1999    disabled GUI stats update to improve performance
//
// 2.0        04-Aug-1999    GUI rework - remove stats and log to separate frames
//                          redesign main panel
//
// 2.80       11-Aug-1999    add freestanding logger frame
//                          add freestanding stats frame
//
// 3.0        19-Aug-99     introduce new file format starting with header
//
/***** PROXY GUI *****/

public class ProxyGUI extends Frame
{
    private static Date lastOCFileDate = null;
    private static OCCache _masterCache = new OCCache();
    private NetListener svr=null;
    private StatsUpdateThread _supdater=null;
    private StatsFrame _statsframe = null;
    private LoggerFrame _loggerframe = null;

    public ProxyGUI()
    {
        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(382,171);
        setVisible(false);
        openFileDialog1.setMode(FileDialog.LOAD);
        openFileDialog1.setTitle("Open");
        //$$ openFileDialog1.move(176,200);
        try {
            borderPanel6.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanel6.setLayout(null);
        add(borderPanel6);
        borderPanel6.setBounds(292,-4,92,168);
        button2.setLabel("Loopback");
        borderPanel6.add(button2);
        button2.setFont(new Font("Dialog", Font.PLAIN, 12));
        button2.setBounds(2,8,68,16);
        button3.setLabel("Whoosh1");

```

```

borderPanel6.add(button3);
button3.setBackground(new java.awt.Color(102,153,153));
button3.setFont(new Font("Dialog", Font.PLAIN, 12));
button3.setBounds(2,24,68,16);
button7.setLabel("Whoosh2");
borderPanel6.add(button7);
button7.setBackground(new java.awt.Color(102,153,153));
button7.setFont(new Font("Dialog", Font.PLAIN, 12));
button7.setBounds(2,40,68,16);
button8.setLabel("Reset");
borderPanel6.add(button8);
button8.setFont(new Font("Dialog", Font.PLAIN, 12));
button8.setBounds(2,56,68,16);
button9.setLabel("Statistics");
borderPanel6.add(button9);
button9.setBounds(2,84,68,16);
button5.setLabel("Clear Stats");
borderPanel6.add(button5);
button5.setFont(new Font("Dialog", Font.PLAIN, 12));
button5.setBounds(2,100,68,16);
button1.setLabel("Logging ");
borderPanel6.add(button1);
button1.setFont(new Font("Dialog", Font.PLAIN, 12));
button1.setBounds(2,124,68,16);
try {
    borderPanel10.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel10.setLayout(null);
add(borderPanel10);
borderPanel10.setBounds(0,-4,296,168);
try {
    borderPanel2.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_NONE);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setLabelColor(java.awt.Color.green);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel2.setLayout(null);
borderPanel10.add(borderPanel2);
borderPanel2.setBackground(java.awt.Color.black);
borderPanel2.setBounds(142,4,136,140);
label11.setText("Bytes Saved");
borderPanel2.add(label11);
label11.setForeground(java.awt.Color.green);
label11.setFont(new Font("Dialog", Font.PLAIN, 14));
label11.setBounds(2,7,96,24);
BwPctLabel.setText("0 %");
borderPanel2.add(BwPctLabel);
BwPctLabel.setForeground(java.awt.Color.green);
BwPctLabel.setFont(new Font("Dialog", Font.PLAIN, 20));
BwPctLabel.setBounds(2,32,112,24);
label12.setText("Packets Saved");
borderPanel2.add(label12);
label12.setForeground(java.awt.Color.green);
label12.setFont(new Font("Dialog", Font.PLAIN, 14));
label12.setBounds(2,68,99,21);
ReqPctLabel.setText("0 %");
borderPanel2.add(ReqPctLabel);
ReqPctLabel.setForeground(java.awt.Color.green);
ReqPctLabel.setFont(new Font("Dialog", Font.PLAIN, 20));
ReqPctLabel.setBounds(2,92,112,24);
try {
    borderPanel5.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_NONE);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel5.setLayout(null);
borderPanel10.add(borderPanel5);

```

```

borderPanel5.setBackground(java.awt.Color.black);
borderPanel5.setBounds(2,4,140,140);
loadTimeLabel.setText("0 s");
borderPanel5.add(loadTimeLabel);
loadTimeLabel.setForeground(java.awt.Color.green);
loadTimeLabel.setFont(new Font("Dialog", Font.PLAIN, 20));
loadTimeLabel.setBounds(-1,32,112,24);
label14.setText("Load Time ");
borderPanel5.add(label14);
label14.setForeground(java.awt.Color.green);
label14.setFont(new Font("Dialog", Font.PLAIN, 14));
label14.setBounds(0,7,108,24);
try {
    borderPanel7.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel7.setLayout(null);
borderPanel5.add(borderPanel7);
borderPanel7.setBounds(-90,140,80,28);
label10.setText("Wwwhoosh!");
borderPanel5.add(label10);
label10.setForeground(java.awt.Color.green);
label10.setFont(new Font("Dialog", Font.ITALIC, 16));
label10.setBounds(-1,68,99,21);
timeLabel.setText("Time:");
borderPanel5.add(timeLabel);
timeLabel.setForeground(java.awt.Color.green);
timeLabel.setBounds(0,94,139,21);
try {
    borderPanel8.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_NONE);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel8.setLayout(null);
add(borderPanel8);
borderPanel8.setBackground(new java.awt.Color(128,0,64));
borderPanel8.setBounds(6,80,124,40);
try {
    borderPanel4.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel4.setLayout(null);
borderPanel8.add(borderPanel4);
borderPanel4.setBounds(-90,140,80,28);
setTitle("Wwwhoosh! Client");
//}}

//{{INIT_MENUS
menu1.setLabel("File");
menu1.add(newMenuItem);
newMenuItem.setLabel("New");
newMenuItem.setShortcut(new MenuShortcut(java.awt.event.KeyEvent.VK_N, false));
menu1.add(openMenuItem);
openMenuItem.setLabel("Open...");
openMenuItem.setShortcut(new MenuShortcut(java.awt.event.KeyEvent.VK_O, false));
menu1.add(saveMenuItem);
saveMenuItem.setLabel("Save");
saveMenuItem.setShortcut(new MenuShortcut(java.awt.event.KeyEvent.VK_S, false));
menu1.add(saveAsMenuItem);
saveAsMenuItem.setLabel("Save As...");
menu1.add(separatorMenuItem);
separatorMenuItem.setLabel("-");
menu1.add(exitMenuItem);
exitMenuItem.setLabel("Exit");
mainMenuBar.add(menu1);
menu4.setLabel("Edit");
menu4.add(menuItem3);
menuItem3.setLabel("Logging Options");
menu4.add(connectionMenuItem);
connectionMenuItem.setLabel("Connection Options");

```

```

menu4.add(menuItem1);
menuItem1.setLabel("Functional Options");
mainMenuBar.add(menu4);
menu2.setLabel("View");
menu2.add(menuItem2);
menuItem2.setLabel("Logging Window");
menu2.add(menuItem4);
menuItem4.setLabel("Statistics Window");
mainMenuBar.add(menu2);
menu3.setLabel("Help");
menu3.add(aboutMenuItem);
aboutMenuItem.setLabel("About...");
mainMenuBar.add(menu3);
//$$ mainMenuBar.move(20,198);
setMenuBar(mainMenuBar);
//}}

//{(REGISTER LISTENERS
SymWindow aSymWindow = new SymWindow();
this.addWindowListener(aSymWindow);
SymAction lSymAction = new SymAction();
openMenuItem.addActionListener(lSymAction);
exitMenuItem.addActionListener(lSymAction);
aboutMenuItem.addActionListener(lSymAction);

button3.addActionListener(lSymAction);
button1.addActionListener(lSymAction);
button2.addActionListener(lSymAction);

menuItem3.addActionListener(lSymAction);
button5.addActionListener(lSymAction);
ConnectionMenuItem.addActionListener(lSymAction);
button7.addActionListener(lSymAction);
button8.addActionListener(lSymAction);
button9.addActionListener(lSymAction);
menuItem1.addActionListener(lSymAction);

menuItem2.addActionListener(lSymAction);
menuItem4.addActionListener(lSymAction);
//}}
try {
    //TODO: Remove hardcoded pathname!
    FileOutputStream bld = new FileOutputStream("ClientBuild.java");
    String verstring = "public class ClientBuild {public static int NUMBER=" +
(ClientBuild.NUMBER+1) + " ; }";
    bld.write(verstring.getBytes());
    bld.close();
}
catch (IOException ioe)
    {System.out.println(ioe); }
}

public ProxyGUI(String title)
{
    this();
    setTitle(title);
}

/**
 * Shows or hides the component depending on the boolean flag b.
 * @param b if true, show the component; otherwise, hide the component.
 * @see java.awt.Component#isVisible
 */
public void setVisible(boolean b)
{
    //Force resize in spite of VCafe
    //setResizable(true);
    if(b)
    {

```

```

        setLocation(50, 50);
    }
    super.setVisible(b);
}

static public void main(String args[])
{
    try
    {
        //Create a new instance of our application's frame, and make it visible.
        ProxyGUI mySvr = new ProxyGUI();
        mySvr.setVisible(true);
        mySvr.init();
        Counters.setLoopback(false);
    }
    catch (Throwable t)
    {
        System.err.println(t);
        t.printStackTrace();
        //Ensure the application exits with an error condition.
        System.exit(1);
    }
}

//
// Update GUI with latest statistics info no more often than limit interval
//
public synchronized void setStats()
{
    //update main panel
    int approx_bytessaved = Counters.OK_304 * 700;
    double pctsaved = (approx_bytessaved * 100.0) / (Counters.getBytesIn() +
Counters.getBytesOut());
    if (pctsaved > 0.0)
    {
        String s1 = new Double(pctsaved).toString() + "      ";
        BwPctLabel.setText(s1.substring(0,5) + " % ");
    }
    double reqsaved = (Counters.OK_304*1000.0) / (Counters.GETS_OUT*10.0);
    if (reqsaved > 0.0)
    {
        String s2 = new Double(reqsaved).toString() + "      ";
        ReqPctLabel.setText( s2.substring(0,5) + " % ");
    }

    double loadt = Counters.getLastEventInterval() / 1000.0;
    String s3 = new Double(loadt).toString() + "      ";
    loadTimeLabel.setText( s3.substring(0,4) + " s" );
    timeLabel.setText(new Date().toGMTString() );

    if (_statsframe == null)
        return;

    if (_statsframe.isShowing())
        _statsframe.setStats();
}

public void init() throws Exception
{
    LocalServiceFactory localFactory = new LocalServiceFactory();
    Debug.setTextArea(_textArea); //set up global ref to logger window
    //logger frame takes its text area from here
    //NO need for this factory stuff in Java!! MLP
    //Pass file/date table to new instances of http proxy service for now
    HttpProxyService httpSvc = new HttpProxyService(_masterCache,

```



```

lastOCFileDate,
this);

localFactory.addService(httpSvc);
Debug.textout.setText("");
Debug.textout.append("Proxy listening on port " + Debug.listenPort +
    " at " + new Date().toGMTString() + "\n");
//
// Start a NetListener.

if (svr != null)
    svr.kill();

svr = new NetListener(Debug.listenPort, localFactory, new HttpTrans());
svr.start();
_supdater = new StatsUpdateThread(this);
_supdater.start();
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    setSize(getInsets().left + getInsets().right + d.width, getInsets().top + getInsets().bottom
+ d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(getInsets().left, getInsets().top);
        components[i].setLocation(p);
    }
    fComponentsAdjusted = true;
}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

long _lastdisplay=0;
java.awt.TextArea _textArea1 = new java.awt.TextArea();

//{{{DECLARE CONTROLS
java.awt.FileDialog openFileDialog1 = new java.awt.FileDialog(this);
symantec.itools.awt.BorderPanel borderPanel6 = new symantec.itools.awt.BorderPanel();
java.awt.Button button2 = new java.awt.Button();
java.awt.Button button3 = new java.awt.Button();
java.awt.Button button7 = new java.awt.Button();
java.awt.Button button8 = new java.awt.Button();
java.awt.Button button9 = new java.awt.Button();
java.awt.Button button5 = new java.awt.Button();
java.awt.Button button1 = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanel10 = new symantec.itools.awt.BorderPanel();
symantec.itools.awt.BorderPanel borderPanel2 = new symantec.itools.awt.BorderPanel();
java.awt.Label label11 = new java.awt.Label();
java.awt.Label BwPctLabel = new java.awt.Label();
java.awt.Label label12 = new java.awt.Label();
java.awt.Label ReqPctLabel = new java.awt.Label();
symantec.itools.awt.BorderPanel borderPanel5 = new symantec.itools.awt.BorderPanel();
java.awt.Label loadTimeLabel = new java.awt.Label();
java.awt.Label label14 = new java.awt.Label();
symantec.itools.awt.BorderPanel borderPanel7 = new symantec.itools.awt.BorderPanel();
java.awt.Label label10 = new java.awt.Label();

```

```

java.awt.Label timeLabel = new java.awt.Label();
symantec.itools.awt.BorderPanel borderPanel8 = new symantec.itools.awt.BorderPanel();
symantec.itools.awt.BorderPanel borderPanel4 = new symantec.itools.awt.BorderPanel();
//}}

```

```

//{{DECLARE_MENUS
java.awt.MenuBar mainMenuBar = new java.awt.MenuBar();
java.awt.Menu menu1 = new java.awt.Menu();
java.awt.MenuItem newMenuItem = new java.awt.MenuItem();
java.awt.MenuItem openMenuItem = new java.awt.MenuItem();
java.awt.MenuItem saveMenuItem = new java.awt.MenuItem();
java.awt.MenuItem saveAsMenuItem = new java.awt.MenuItem();
java.awt.MenuItem separatorMenuItem = new java.awt.MenuItem();
java.awt.MenuItem exitMenuItem = new java.awt.MenuItem();
java.awt.Menu menu4 = new java.awt.Menu();
java.awt.MenuItem menuItem3 = new java.awt.MenuItem();
java.awt.MenuItem ConnectionMenuItem = new java.awt.MenuItem();
java.awt.MenuItem menuItem1 = new java.awt.MenuItem();
java.awt.Menu menu2 = new java.awt.Menu();
java.awt.MenuItem menuItem2 = new java.awt.MenuItem();
java.awt.MenuItem menuItem4 = new java.awt.MenuItem();
java.awt.Menu menu3 = new java.awt.Menu();
java.awt.MenuItem aboutMenuItem = new java.awt.MenuItem();
//}}

```

```

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == ProxyGUI.this)
            Frame1_WindowClosing(event);
    }
}

```

```

void Frame1_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    Frame1_WindowClosing_Interaction1(event);
}

```

```

void Frame1_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        // QuitDialog Create and show as modal
        (new QuitDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

```

```

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == openMenuItem)
            openMenuItem_ActionPerformed(event);
        else if (object == aboutMenuItem)
            aboutMenuItem_ActionPerformed(event);
        else if (object == exitMenuItem)
            exitMenuItem_ActionPerformed(event);
        else if (object == button3)
            button3_ActionPerformed(event);
        if (object == button1)
            button1_ActionPerformed(event);
    }
}

```

```

        else if (object == button2)
            button2_ActionPerformed(event);
//
//        else if (object == button4)
//            button4_ActionPerformed(event);
        else if (object == menuItem3)
            menuItem3_ActionPerformed(event);
        else if (object == button5)
            button5_ActionPerformed(event);
//
//        else if (object == button6)
//            button6_ActionPerformed(event);
        else if (object == ConnectionMenuItem)
            ConnectionMenuItem_ActionPerformed(event);
        else if (object == button7)
            button7_ActionPerformed(event);
        else if (object == button8)
            button8_ActionPerformed(event);
        else if (object == menuItem1)
            menuItem1_ActionPerformed(event);
        else if (object == button9)
            button9_ActionPerformed(event);
        else if (object == menuItem2)
            menuItem2_ActionPerformed(event);
        else if (object == menuItem4)
            menuItem4_ActionPerformed(event);
    }
}

void openMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    openMenuItem_ActionPerformed_Interaction1(event);
}

void openMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // OpenFileDialog Create and show as modal
        int      defMode      = openFileDialog1.getMode();
        String    defTitle     = openFileDialog1.getTitle();
        String    defDirectory = openFileDialog1.getDirectory();
        String    defFile      = openFileDialog1.getFile();

        openFileDialog1 = new java.awt.FileDialog(this, defTitle, defMode);
        openFileDialog1.setDirectory(defDirectory);
        openFileDialog1.setFile(defFile);
        openFileDialog1.setVisible(true);
    } catch (Exception e) {
    }
}

void aboutMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    aboutMenuItem_ActionPerformed_Interaction1(event);
}

void exitMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    exitMenuItem_ActionPerformed_Interaction1(event);
}

void exitMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{

```

```

    try {
        // QuitDialog Create and show as modal
        (new QuitDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

void button3_ActionPerformed(java.awt.event.ActionEvent event)
{
    if (!Debug.PASSTHRU )
    {
        Debug.PASSTHRU=true;

        button3.setBackground(Color.lightGray);

        //Make sure OCD processing is shut off too since it depend on OC

        button7.setBackground(Color.lightGray);
        Debug.WWWhoosh2=false;
        //java.awt.Toolkit tk = new java.awt.Toolkit();
        //tk.beep();
    }
    else
    {
        button3.setBackground(new Color(102,153,153) );
        Debug.PASSTHRU=false;
    }
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    if (Debug.TTOUT)
    {
        Debug.off();
        button1.setBackground(Color.lightGray );
        Debug.forceprint("\n" + "Logging stopped at " + new Date().toGMTString() );
    }
    else
    {
        Debug.on();
        //button1.setBackground(new Color( 102,153,153 ));
        Debug.forceprint("\n" + "Logging started at " + new Date().toGMTString() + "\n" );
    }
    button1_ActionPerformed_Interaction1(event);
}

void button2_ActionPerformed(java.awt.event.ActionEvent event)
{
    if (Counters.getLoopback())
    {
        Counters.setLoopback(false);
        button2.setBackground(Color.lightGray);
    }
    else
    {
        Counters.setLoopback(true);
        button2.setBackground(new Color(102,153,153) );
    }
}

```

```
void button4_ActionPerformed(java.awt.event.ActionEvent event)
{
    Debug.textout.setText(" ");
}

void Lo_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    Lo_ActionPerformed_Interaction1(event);
}

void Lo_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // LogLevelDialog Create and show as non-modal
        (new LogLevelDialog(this, false)).setVisible(true);
    } catch (Exception e) {
    }
}

void menuItem3_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    menuItem3_ActionPerformed_Interaction1(event);
}

void menuItem3_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // LogLevelDialog Create and show as modal
        (new LogLevelDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

void button5_ActionPerformed(java.awt.event.ActionEvent event)
{
    Counters.reset();
    setStats();
    BwPctLabel.setText("0 %");
    loadTimeLabel.setText("0 s");
    ReqPctLabel.setText("0 %");
}

void aboutMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // AboutDialog Create and show as non-modal
        (new AboutClientDialog(this, false)).setVisible(true);
    } catch (Exception e) {
    }
}

void ConnectionMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    ConnectionMenuItem_ActionPerformed_Interaction1(event);
}

void ConnectionMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // ConnectionDialog Create and show as modal
        (new ConnectionDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}
```

```

    }
}

void button7_ActionPerformed(java.awt.event.ActionEvent event)
{
    if ( (Debug.WWWhoosh2==false) && ( !Debug.PASSTHRU ) )
    {
        Debug.WWWhoosh2=true;

        button7.setBackground( new Color(102,153,153) );
        //java.awt.Toolkit tk = new java.awt.Toolkit();
        //tk.beep();
    }
    else
    {
        button7.setBackground(Color.lightGray);
        Debug.WWWhoosh2=false;
    }
}

void button8_ActionPerformed(java.awt.event.ActionEvent event)
{
    _masterCache.clearCaches();
    Debug.print(1,"CACHE CONTENTS ERASED" );
    //_statswin.toFront();
}

void menuItem1_ActionPerformed(java.awt.event.ActionEvent event)
{
    menuItem1_ActionPerformed_Interaction1(event);
}

void menuItem1_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // FeatureDialog Create and show as modal
        (new FeatureDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

void button1_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    if ( (_loggerframe != null) && ( _loggerframe.isShowing() ) )
        return;

    try {
        // LoggerFrame Create and show the LoggerFrame
        _loggerframe = new LoggerFrame();
        _loggerframe.setVisible(true);
        // button1.setBackground(new Color( 102,153,153 ));
    }
    catch (Exception e) { }
}

void button9_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    button9_ActionPerformed_Interaction1(event);
}

void button9_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    if ( (_statsframe != null) && ( _statsframe.isShowing() ) )

```

```

    {
        _statsframe.hide();
        return;
    };

    try {
        // StatsFrame Create and show the StatsFrame
        _statsframe = new StatsFrame();
        _statsframe.setVisible(true);
    } catch (Exception e) {
    }
}

void menuItem2_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    menuItem2_ActionPerformed_Interaction1(event);
}

void menuItem4_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    menuItem4_ActionPerformed_Interaction1(event);
}

void menuItem4_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // StatsFrame Create and show the StatsFrame
        _statsframe = new StatsFrame();
        _statsframe.setVisible(true);
    } catch (Exception e) {
    }
}

void menuItem2_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    if ( (_loggerframe != null) && ( _loggerframe.isShowing() ) )
        return;

    try {
        // LoggerFrame Create and show the LoggerFrame
        _loggerframe = new LoggerFrame();
        _loggerframe.setVisible(true);
        //button1.setBackground(new Color( 102,153,153 ));
    }
    catch (Exception e) { }
}
}

```

```

public class ProxyGUIBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //{{Start Automatic Code Generation Block
        {"openFileDialog1_title", "Open"},
        {"openFileDialog1_directory", ""},
        {"openFileDialog1_file", ""},
        {"button2_label", "Loopback: OFF"},
        {"button3_label", "WWWwhoosh1: ON"},
        {"button4_label", "Clear Log"},
        {"button1_label", "Logging: OFF"},
        {"borderPanell_label", ""},
        {"label1_text", "HTTP Requests:"},
        {"label8_text", "HTTP Forwarded:"},
        {"getsInLabel_text", "000"},
        {"getsOutLabel_text", "000"},
        {"label3_text", "GET ms:"},
        {"RTmsLabel_text", "000"},
        {"bytesSentLabel_text", "000"},
        {"bytesRxLabel_text", "000"},
        {"label4_text", "Bytes sent:"},
        {"label5_text", "Bytes received:"},
        {"label2_text", "HTTP Resolved:"},
        {"ACKLabel_text", "000"},
        {"eventElapsedLabel_text", "000"},
        {"label6_text", "WWWwhoosh-2 read: "},
        {"WWWwhoosh2byteslabel_text", "000"},
        {"label7_text", "Load Time ms"},
        {"textAreal_text", ""},
        {"button5_label", "Clear Stats"},
        {"button6_label", "File out: OFF"},
        {"button7_label", "WWWwhoosh2: ON"},
        {"button8_label", "Reset Caches"},
        {"button9_label", "Statistics"},
        {"ProxyGUI_title", "WWWwhoosh Proxy"},
        {"menu1_label", "File"},
        {"newMenuItem_label", "New"},
        {"openMenuItem_label", "Open..."},
        {"saveMenuItem_label", "Save"},
        {"saveAsMenuItem_label", "Save As..."},
        {"separatorMenuItem_label", "-"},
        {"exitMenuItem_label", "Exit"},
        {"menu4_label", "Edit"},
        {"menuItem3_label", "Logging Options"},
        {"ConnectionMenuItem_label", "Connection Options"},
        {"menuItem1_label", "Functional Options"},
        {"menu3_label", "Help"},
        {"aboutMenuItem_label", "About..."},
        //}}End Automatic Code Generation Block
    };
}

```



```

/*
 * A basic extension of the java.awt.Dialog class
 */

import java.awt.*;
import java.awt.event.*;
import symantec.itools.awt.BorderPanel;

public class QuitDialog extends Dialog
{
    public QuitDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        //Keep a local reference to the invoking frame
        frame = parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(246,130);
        setVisible(false);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(7,6,239,95);
        labell.setText("Do you really want to exit?");
        labell.setAlignment(java.awt.Label.CENTER);
        borderPanell.add(labell);
        labell.setBounds(22,26,168,23);
        noButton.setLabel(" No ");
        add(noButton);
        noButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        noButton.setBounds(170,101,64,20);
        yesButton.setLabel(" Yes ");
        add(yesButton);
        yesButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        yesButton.setBounds(104,101,64,20);
        setTitle("Exit WWWhoosh Client");
        //}}}

        //{{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        noButton.addActionListener(lSymAction);
        yesButton.addActionListener(lSymAction);
        //}}}
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;

        // Adjust components according to the insets

```

```

        setSize(getInsets().left + getInsets().right + d.width, getInsets().top + getInsets().bottom
+ d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(getInsets().left, getInsets().top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

    public QuitDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    /**
     * Shows or hides the component depending on the boolean flag b.
     * @param b if true, show the component; otherwise, hide the component.
     * @see java.awt.Component#isVisible
     */
    public void setVisible(boolean b)
    {
        if(b)
        {
            Rectangle bounds = getParent().getBounds();
            Rectangle abounds = getBounds();

            setLocation(bounds.x + (bounds.width - abounds.width) / 2,
                bounds.y + (bounds.height - abounds.height) / 2);
            Toolkit.getDefaultToolkit().beep();
        }
        super.setVisible(b);
    }

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;
    // Invoking frame
    Frame frame = null;

    /**{DECLARE_CONTROLS
    symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
    java.awt.Label labell = new java.awt.Label();
    java.awt.Button noButton = new java.awt.Button();
    java.awt.Button yesButton = new java.awt.Button();
    /**}

    class SymAction implements java.awt.event.ActionListener
    {
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
            Object object = event.getSource();
            if (object == yesButton)
                yesButton_ActionPerformed(event);
            else if (object == noButton)
                noButton_ActionPerformed(event);
        }
    }

    void yesButton_ActionPerformed(java.awt.event.ActionEvent event)
    {
        // to do: code goes here.

        yesButton_ActionPerformed_Interaction1(event);
    }

```

```

void yesButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        frame.setVisible(false);    // Hide the invoking frame
        frame.dispose();            // Free system resources
        this.dispose();             // Free system resources
        System.exit(0);             // close the application
    } catch (Exception e) {
    }
}

void noButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    noButton_ActionPerformed_Interaction1(event);
}

void noButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == QuitDialog.this)
            QuitDialog_WindowClosing(event);
    }
}

void QuitDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    QuitDialog_WindowClosing_Interaction1(event);
}

void QuitDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

static java.util.ResourceBundle quitDialogBundle =
java.util.ResourceBundle.getBundle("QuitDialogBundle");

```

```
public class QuitDialogBundle extends java.util.ListResourceBundle
{
    public Object[][] getContents()
    {
        return contents;
    }

    static final Object[][] contents =
    {
        //Changing tags in the following block will
        //break the parsing of the related file
        //{{Start Automatic Code Generation Block
        {"yesButton_label", " Yes "},
        {"noButton_label", " No "},
        {"label1_text", "Do you really want to exit?"},
        {"QuitDialog_title", "Exit WWWhoosh Client"},
        //}}End Automatic Code Generation Block
    };
}
```

```

/*
  A basic extension of the java.awt.Frame class
*/

import java.awt.*;
import symantec.itools.awt.BorderPanel;
import COM.reitshamer.util.Counters;

public class StatsFrame extends Frame
{
    public StatsFrame()
    {
        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //({INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(373,187);
        setVisible(false);
        button1.setLabel("Close");
        add(button1);
        button1.setBounds(289,158,64,20);
        button2.setLabel("Clear");
        add(button2);
        button2.setBounds(225,158,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setIPadBottom(4);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setLabel("Statistics");
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingRight(4);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingBottom(4);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingTop(4);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingLeft(4);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(7,12,359,143);
        RTmsLabel.setText("000");
        borderPanell.add(RTmsLabel);
        RTmsLabel.setBounds(300,20,72,16);
        label1.setText("HTTP Requests:");
        borderPanell.add(label1);
    }
}

```

```

label1.setFont(new Font("Dialog", Font.PLAIN, 12));
label1.setBounds(6, 20, 124, 16);
label8.setText("HTTP Forwarded:");
borderPanel1.add(label8);
label8.setFont(new Font("Dialog", Font.PLAIN, 12));
label8.setBounds(6, 35, 124, 16);
getsInLabel.setText("000");
borderPanel1.add(getsInLabel);
getsInLabel.setBounds(130, 19, 72, 16);
getsOutLabel.setText("000");
borderPanel1.add(getsOutLabel);
getsOutLabel.setBounds(130, 34, 72, 16);
label13.setText("GET ms:");
borderPanel1.add(label13);
label13.setFont(new Font("Dialog", Font.PLAIN, 12));
label13.setBounds(213, 20, 84, 16);
bytesSentLabel.setText("000");
borderPanel1.add(bytesSentLabel);
bytesSentLabel.setBounds(300, 35, 72, 16);
bytesRxLabel.setText("000");
borderPanel1.add(bytesRxLabel);
bytesRxLabel.setBounds(130, 64, 72, 16);
label4.setText("Bytes Sent:");
borderPanel1.add(label4);
label4.setFont(new Font("Dialog", Font.PLAIN, 12));
label4.setBounds(213, 35, 84, 16);
label5.setText("Bytes Received:");
borderPanel1.add(label5);
label5.setFont(new Font("Dialog", Font.PLAIN, 12));
label5.setBounds(6, 64, 120, 14);
label2.setText("HTTP Resolved:");
borderPanel1.add(label2);
label2.setFont(new Font("Dialog", Font.PLAIN, 12));
label2.setBounds(6, 49, 124, 16);
ACKLabel.setText("000");
borderPanel1.add(ACKLabel);
ACKLabel.setBounds(130, 49, 72, 16);
eventElapsedLabel.setText("000");
borderPanel1.add(eventElapsedLabel);
eventElapsedLabel.setBounds(130, 80, 74, 15);
label6.setText("WWWhoosh-2 read: ");
borderPanel1.add(label6);
label6.setBounds(213, 64, 83, 17);
WWWhoosh2bytesLabel.setText("000");
borderPanel1.add(WWWhoosh2bytesLabel);
WWWhoosh2bytesLabel.setBounds(300, 64, 72, 16);
label7.setText("Load Time ms");
borderPanel1.add(label7);
label7.setBounds(6, 79, 122, 16);
label9.setText("Bytes Saved");
borderPanel1.add(label9);
label9.setBounds(213, 51, 84, 14);
bytesSavedLabel.setText("000");
borderPanel1.add(bytesSavedLabel);
bytesSavedLabel.setBounds(300, 49, 72, 16);
setResizable(false);
//}}

//{{{INIT_MENU
//}}

//{{{REGISTER_LISTENERS
SymWindow aSymWindow = new SymWindow();
this.addWindowListener(aSymWindow);
SymAction lSymAction = new SymAction();
button1.addActionListener(lSymAction);
button2.addActionListener(lSymAction);
//}}
}

```

```

public StatsFrame(String title)
{
    this();
    setTitle(title);
}

/**
 * Shows or hides the component depending on the boolean flag b.
 * @param b if true, show the component; otherwise, hide the component.
 * @see java.awt.Component#isVisible
 */
public void setVisible(boolean b)
{
    //Force no-resize in spite of VCafe
    setResizable(false);
    if(b)
    {
        setLocation(450, 50);
    }
    super.setVisible(b);
}

static public void main(String args[])
{
    (new StatsFrame()).setVisible(true);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }
    fComponentsAdjusted = true;
}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

//{{{DECLARE_CONTROLS
java.awt.Button button1 = new java.awt.Button();
java.awt.Button button2 = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
java.awt.Label RTmsLabel = new java.awt.Label();
java.awt.Label label1 = new java.awt.Label();
java.awt.Label label8 = new java.awt.Label();
java.awt.Label getsInLabel = new java.awt.Label();
java.awt.Label getsOutLabel = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
java.awt.Label bytesSentLabel = new java.awt.Label();
java.awt.Label bytesRxLabel = new java.awt.Label();
java.awt.Label label4 = new java.awt.Label();
java.awt.Label label5 = new java.awt.Label();

```

```

java.awt.Label label2 = new java.awt.Label();
java.awt.Label ACKLabel = new java.awt.Label();
java.awt.Label eventElapsedLabel = new java.awt.Label();
java.awt.Label label6 = new java.awt.Label();
java.awt.Label WWWhoosh2byteslabel = new java.awt.Label();
java.awt.Label label7 = new java.awt.Label();
java.awt.Label label9 = new java.awt.Label();
java.awt.Label bytesSavedLabel = new java.awt.Label();
//}}

//{{DECLARE_MENUS
//}}

//
// Update GUI with latest statistics info no more often than limit interval
//
public synchronized void setStats()
{
    //passive method of throttling GUI display
    //update main panel
    int approx_bytessaved = Counters.OK_304 * 700;
    bytesSavedLabel.setText(new Integer(approx_bytessaved).toString());
    getsInLabel.setText(new Integer(Counters.GETS_IN).toString());
    getsOutLabel.setText(new Integer(Counters.GETS_OUT).toString());
    ACKLabel.setText(new Integer(Counters.OK_304).toString());
    bytesRxLabel.setText(new Long(Counters.getBytesIn()).toString());
    WWWhoosh2byteslabel.setText(new Long(Counters.OCDbytesIn).toString());
    eventElapsedLabel.setText(new Long(Counters.getLastEventInterval()).toString());
    bytesSentLabel.setText(new Long(Counters.getBytesOut()).toString());
    bytesSavedLabel.setText(new Long(approx_bytessaved).toString());
    if (Counters.GETms() >= 0)
        RTmsLabel.setText(new Long(Counters.GETms()).toString());
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == StatsFrame.this)
            StatsFrame_WindowClosing(event);
    }
}

void StatsFrame_WindowClosing(java.awt.event.WindowEvent event)
{
    setVisible(false);    // hide the Frame
}

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == button1)
            button1_ActionPerformed(event);
        else if (object == button2)
            button2_ActionPerformed(event);
    }
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    button1_ActionPerformed_Interaction1(event);
}

void button1_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)

```



```
{
    try {
        // StatsFrame Hide the StatsFrame
        this.setVisible(false);
    } catch (Exception e) {
    }
}

void button2_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    Counters.reset();
    setStats();
}
}
```

```

import java.awt.*;

public class StatsHelpDialog extends Dialog {
    public StatsHelpDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //{{{INIT_CONTROLS
        setLayout(null);
        setFont(new Font("Dialog", Font.PLAIN, 14));
        setSize(351,185);
        setVisible(false);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setBackground(java.awt.Color.lightGray);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 14));
        okButton.setBounds(140,128,60,24);
        label1.setText("GETs out: counts GET requests forwarded to server");
        add(label1);
        label1.setFont(new Font("Dialog", Font.PLAIN, 12));
        label1.setBounds(12,40,420,36);
        label2.setText("304s back: counts GET requests satisfied via proxy ");
        add(label2);
        label2.setFont(new Font("Dialog", Font.PLAIN, 12));
        label2.setBounds(12,68,420,36);
        label3.setText("GETs in: counts HTTP GET requests from browser");
        add(label3);
        label3.setFont(new Font("Dialog", Font.PLAIN, 12));
        label3.setBounds(12,12,420,36);
        setTitle("ICCS statistics");
        //}}}

        //{{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        okButton.addActionListener(lSymAction);
        //}}}
    }

    public StatsHelpDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        // Only do this once.
        if (fComponentsAdjusted)

```

```

        return;

// Adjust components according to the insets
Insets insets = getInsets();
setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
Component components[] = getComponents();
for (int i = 0; i < components.length; i++)
{
    Point p = components[i].getLocation();
    p.translate(insets.left, insets.top);
    components[i].setLocation(p);
}

// Used for addNotify check.
fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }

    super.setVisible(b);
}

//{{{DECLARE_CONTROLS
java.awt.Button okButton = new java.awt.Button();
java.awt.Label label1 = new java.awt.Label();
java.awt.Label label2 = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
//}}}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter

```

```
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == StatsHelpDialog.this)
            StatsHelpDialog_WindowClosing(event);
    }
}

void StatsHelpDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    StatsHelpDialog_WindowClosing_Interaction1(event);
}

void StatsHelpDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}
}
```

```
import java.awt.*;
```

```
public class StatsHelpDialog extends Dialog {
    public StatsHelpDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //({{INIT_CONTROLS
        setLayout(null);
        setFont(new Font("Dialog", Font.PLAIN, 14));
        setSize(351,185);
        setVisible(false);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setBackground(java.awt.Color.lightGray);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 14));
        okButton.setBounds(140,128,60,24);
        label1.setText("GETs out: counts GET requests forwarded to server");
        add(label1);
        label1.setFont(new Font("Dialog", Font.PLAIN, 12));
        label1.setBounds(12,40,420,36);
        label2.setText("304s back: counts GET requests satisfied via proxy ");
        add(label2);
        label2.setFont(new Font("Dialog", Font.PLAIN, 12));
        label2.setBounds(12,68,420,36);
        label3.setText("GETs in: counts HTTP GET requests from browser");
        add(label3);
        label3.setFont(new Font("Dialog", Font.PLAIN, 12));
        label3.setBounds(12,12,420,36);
        setTitle("ICCS statistics");
        //}}

        //({{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        okButton.addActionListener(lSymAction);
        //}}

    }

    public StatsHelpDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        // Only do this once.
        if (fComponentsAdjusted)
```

```

        return;

// Adjust components according to the insets
Insets insets = getInsets();
setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
Component components[] = getComponents();
for (int i = 0; i < components.length; i++)
{
    Point p = components[i].getLocation();
    p.translate(insets.left, insets.top);
    components[i].setLocation(p);
}

// Used for addNotify check.
fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }

    super.setVisible(b);
}

//{{{DECLARE_CONTROLS
java.awt.Button okButton = new java.awt.Button();
java.awt.Label label1 = new java.awt.Label();
java.awt.Label label2 = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
//}}}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter

```

```
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == StatsHelpDialog.this)
            StatsHelpDialog_WindowClosing(event);
    }
}

void StatsHelpDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    StatsHelpDialog_WindowClosing_Interaction1(event);
}

void StatsHelpDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}
}
```

```

import java.net.*;
import java.io.*;
import COM.reitshamer.http.OCDResponse;
import COM.reitshamer.error.*;
import COM.reitshamer.util.*;
import COM.reitshamer.dc.*;
/***** StatsUpdateThread *****/
//
// @author Mark Palmer
// @version 0.1 - initial beta release, 1-Aug-1999
//
// Modifications:
//
//
/***** OCDReaderThread *****/
/*
 * @input _sock      Socket that the incoming request is coming in on
 * @input ocdR - the OCD Response to be processed via this socket
 */
public class StatsUpdateThread extends Thread
{
    private ProxyGUI _pgui=null;

    public StatsUpdateThread(ProxyGUI pgui)
    {
        super();
        _pgui = pgui;
        //{{{ INIT_CONTROLS
        //}}
    }

    public void run()
    {
        while (true)
        {
            _pgui.setStats();
            try
            {
                sleep(250);
            }
            catch (InterruptedException ie)
            {
                {}
            }
        }
        //{{{ DECLARE_CONTROLS
        //}}
    }
}

```



```

/*
 * A basic extension of the java.awt.Dialog class
 */

import java.awt.*;
import java.awt.image.ImageObserver;
import symantec.itools.awt.BorderPanel;
import symantec.itools.awt.InvisibleHTMLLink;
import symantec.itools.awt.InvisibleButton;

public class AboutServerDialog extends Dialog {

    public AboutServerDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setFont(new Font("Dialog", Font.PLAIN, 12));
        setSize(338,231);
        setVisible(false);
        try {
            borderPanel1.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanel1.setLayout(null);
        add(borderPanel1);
        borderPanel1.setBounds(8,8,327,193);
        label4.setText("Patents Pending");
        borderPanel1.add(label4);
        label4.setBounds(26,78,156,24);
        label3.setText("All Rights Reserved");
        borderPanel1.add(label3);
        label3.setBounds(26,60,156,24);
        label2.setText("Copyright (C) 1999 MTI");
        borderPanel1.add(label2);
        label2.setBounds(26,36,156,24);
        label5.setText("Customer:");
        borderPanel1.add(label5);
        label5.setBounds(26,116,78,20);
        label6.setText("Version: 2");
        borderPanel1.add(label6);
        label6.setBounds(26,135,76,15);
        label7.setText("Build: ");
        borderPanel1.add(label7);
        label7.setBounds(26,150,41,21);
        borderPanel1.add(textField1);
        textField1.setBounds(102,117,190,17);
        numlabel.setText("0");
        borderPanel1.add(numlabel);
        numlabel.setBounds(77,151,51,21);
        button1.setLabel("WWWhoosh! Site Optimizer");
        borderPanel1.add(button1);
        button1.setBounds(22,14,181,21);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        okButton.setBounds(262,203,64,20);
        setTitle("About WWWhoosh Site Optimizer");
        //}}}

        numlabel.setText((new Integer(Build.NUMBER)).toString() );
    }
}

```

```

    //({REGISTER_LISTENERS
    SymWindow aSymWindow = new SymWindow();
    this.addWindowListener(aSymWindow);
    SymAction lSymAction = new SymAction();
    okButton.addActionListener(lSymAction);
    button1.addActionListener(lSymAction);
    //})

}

public AboutServerDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom +
d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

public void paint (Graphics g)
{
    //Image logo2 = getToolkit().getImage("IOP.gif");
    Image logo = getToolkit().getImage("Javalogo.jpg");
    //g.drawImage(logo2,45,50,200,75,(ImageObserver)this);
    g.drawImage(logo,15,35,65,65,(ImageObserver)this);
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width)/ 2,
            bounds.y + (bounds.height - abounds.height)/2);
    }

    super.setVisible(b);
}

//({DECLARE_CONTROLS
symantec.itools.awt.BorderPanel borderPanel1 = new symantec.itools.awt.BorderPanel();
java.awt.Label label4 = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
java.awt.Label label2 = new java.awt.Label();

```

```

java.awt.Label label5 = new java.awt.Label();
java.awt.Label label6 = new java.awt.Label();
java.awt.Label label7 = new java.awt.Label();
java.awt.TextField textField1 = new java.awt.TextField();
java.awt.Label numlabel = new java.awt.Label();
java.awt.Button button1 = new java.awt.Button();
java.awt.Button okButton = new java.awt.Button();
//}}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
        else if (object == button1)
            button1_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == AboutServerDialog.this)
            AboutServerDialog_WindowClosing(event);
    }
}

void AboutServerDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.
    AboutServerDialog_WindowClosing_Interaction1(event);
}

void AboutServerDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)

```

```
{  
    Runtime rt = Runtime.getRuntime();  
    try {  
        rt.exec("C:\\Program Files\\Internet Explorer\\iexplore.exe  
http://www.whoosh.com");  
    }  
    catch (java.io.IOException ioe)  
    {}  
}
```

```
public class Build {public static int VERSION=2; public static int NUMBER=1007 ; }
```

```

import java.io.*;
import java.util.Date;

/***** CONFIG *****/
//
// The Config class is responsible for governing the runtime operation of the Optimizer
// by the settings of its various parameters - whether to user compression, whether to
// generate OCD information, what file extensions to process, etc.
// It also writes and restores itself from an ASCII configuration file.
//
/***** CONFIG *****/
public class Config
{
    // The following function is a placeholder for control initialization.
    // You should call this function from a constructor or initialization function.
    public void vcInit() {
        //{{{ INIT CONTROLS
        //$$ config1.move(0,0);
        //}}}
    }

    //hardcode version until it changes!
    public static int VERSION = 2;

    private static PrintStream outStream = System.out;
    private static boolean _modified=false;
    private static boolean _GZIP = true;
    private static boolean _OCD1 = true;
    private static boolean _continuous=false;
    private static int _geninterval=1;
    private static int _numprocessed=0;
    private static String _curFile=null;
    private static String _curOutFile=null;
    private static String _curDir="C:";
    private static String _rootDir= "C:/temp/yourdomaindirectory";
    private static String _configFName = "WWWhoosh.INI";
    private static String _logfilename="WWWhooshServerLog.txt";
    private static int _maxDepth=10;
    private static int _curDepth=0;
    private static String _extensionList = "htm html";
    private static String _subobjectExtensions = "gif jpg jpeg";
    private static String _defaultFile="default.htm";
    private static String _domain="www.yourdomainhere.com";
    private static String _dateStamp="date last run";
    private static String[] _fileList = {"default.htm"};
    private static long _globalRunTime=0;

    public static void setRunTime()
    { _globalRunTime = System.currentTimeMillis(); }

    public static long getRunTime()
    { return _globalRunTime; }

    public static boolean getContinuous()
    { return _continuous; }

    public static void setContinuous(boolean c)
    { _continuous = c; }

    public static int getGenInterval()
    { return _geninterval; }

    public static void setGenInterval(int g)
    { _geninterval=g; }

    public static String getDate()
    { return _dateStamp; }
    public static void setDate(String d)
    { _dateStamp = d; _modified=true; }

```

```

public static String getDefaultFile()
{return _defaultFile;}
public static void setDefaultFile(String d)
{ _defaultFile = d; _modified=true; }

public static String getRootDir()
{return _rootDir;}

public static void setRootDir(String d)
{ _rootDir = d.replace('\\', '/');
  _rootDir = _rootDir.toLowerCase();
  if (d == null)
    _rootDir = System.getProperty("user.dir");
  _modified=true;
  if (!_rootDir.endsWith("/"))
    _rootDir += "/";
}

public static String getCurInFile()
{return _curFile;}
public static void setCurInFile(String d)
{ _curFile = d;}

public static String getCurOutFile()
{return _curOutFile;}
public static void setCurOutFile(String d)
{ _curOutFile = d.toLowerCase(); }

public static void setGZip(boolean tf)
{ _GZIP = tf;}
public static boolean getGZip()
{return _GZIP;}

public static void setOCD1(boolean tf)
{ _OCD1 = tf;}
public static boolean getOCD1()
{return _OCD1;}

public static boolean modified()
{return _modified;}
public static void setModified(boolean m)
{ _modified = m;}

public static String getFile()
{return _configFName;}
public static void setFile(String f)
{ _configFName = f;
  File nf = new File(f);
  _rootDir = nf.getParent();
  _modified=true;}

public static void setLogFile(String lfname)
{ _logfile = lfname;}
public static String getLogFile()
{return _logfile;}

public static void setFiles(String[] s)
{ _fileList = s; _modified=true; }
public static String[] getFiles()
{ return _fileList; }

public static void setDomain(String s)
{ _domain = s.replace('\\', '/');
  _domain = _domain.toLowerCase();
  _modified=true; }
public static String getDomain()
{return _domain;}

```

```

public static void setSubobjectExtensions(String s)
{
    _subobjectExtensions = s.toLowerCase(); _modified=true; }
public static String getSubobjectExtensions()
{
    return _subobjectExtensions;
}

public static void setExtensions(String s)
{
    _extensionList = s.toLowerCase(); _modified=true; }
public static String getExtensions()
{
    return _extensionList;
}

//general rule for detecting filenames: if valid extension after last dot
public static boolean isSubobject(String f)
{
    if (f== null)
        return false;

    String filename = f.toLowerCase();
    int dot = filename.lastIndexOf(".gif");
    //if (dot > -1)
    //    OutMgr.print("      GIF: " + filename + "\n");

    dot = filename.lastIndexOf(".");
    if (dot == -1)
        return false;

    if ( (filename.length() < dot+4) || (filename.length() > dot+5))
        return false;

    String ext = filename.substring(dot+1, dot+4) ;

    if ( _subobjectExtensions.indexOf(ext.toLowerCase()) > -1)
    {
        //    OutMgr.print("      FOUND SUBOBJECT " + filename );

        return true;
    }
    else
        return false;
}

/***** DETECT HTML EXTENSIONS *****/
public static boolean inExtensionList(String ext)
{
    //TODO: FIX - breaks for extensions using mixed case e.g. ".Htm"
    if ( (_extensionList.indexOf(ext.toUpperCase()) > -1)
        || (_extensionList.indexOf(ext.toLowerCase()) > -1) )
        return true;
    else
        return false;
}

/***** DETECT SUBOBJECT FILE EXTENSIONS *****/
public static boolean inSubobjectExtensionList(String ext)
{
    String tmp = ext.toLowerCase();

    if ( _subobjectExtensions.indexOf(tmp) != -1)
        return true;
    else
        return false;
}

/***** LOAD LAST USED CONFIG *****/
public static boolean loadLast()
{
    //get a copy of the last-used config from client.cfg,
    //then proceed to load the actual last-used config

```

```

loadFrom("WWWhoosh.INI"); //bootstrap
if (!_configFName.equalsIgnoreCase("WWWhoosh.INI"))
    loadFrom(_configFName);

return true;
}

/*****\***** LOAD FROM FILE
*****/
public static boolean loadFrom(String fn)
{
    File fi = new File(fn);
    if (!fi.exists())
    {
        System.out.println("**** WARNING: configuration file not found ****"
            + " creating new" + OutMgr.chopPath(fi.getAbsolutePath()) );
        saveAs("WWWhoosh.INI"); //saves in default run directory

        //_rootDir=fi.getParent();

        return false;
    }
    try
    {
        FileInputStream fin = new FileInputStream(fi);
        DataInputStream din = new DataInputStream(fin);

        _rootDir=fi.getParent(); //config file directory is used as web file root for
now!

        String tagline = din.readLine();
        String val=null;

        while (tagline != null)
        {
            val = tagline.substring(tagline.indexOf("=")+1).trim();
            if (tagline.startsWith("DateLastRun"))
            {
                Config.setDate(val);
            }
            if (tagline.startsWith("NonameFile"))
            {
                val = tagline.substring(tagline.indexOf("=")+1).trim();
                Config.setDefaultFile(val);
            }
            if (tagline.startsWith("RootDir"))
            {
                val = tagline.substring(tagline.indexOf("=")+1).trim();
                setRootDir(val);
            }

            if (tagline.startsWith("Domain"))
            {
                val = tagline.substring(tagline.indexOf("=")+1).trim();
                Config.setDomain(val);
            }
            if (tagline.startsWith("LastConfigFile"))
            {
                val = tagline.substring(tagline.indexOf("=")+1).trim();
                _configFName = val;
            }
            if (tagline.startsWith("LogFile"))
            {
                val = tagline.substring(tagline.indexOf("=")+1).trim();
                _logfilename = val;
                OutMgr.setLogfile(val);
            }
            if (tagline.startsWith("FILEOUT"))
            {
                val = tagline.substring(tagline.indexOf("=")+1).trim();

```



```

        if (val == "true")
            OutMgr.setLogOn();
        else
            OutMgr.setLogOff();
    }

    if (tagline.startsWith("LOGAPPEND"))
    {
        val = tagline.substring(tagline.indexOf("=")+1).trim();

        if (val == "true")
            OutMgr.setLogAppend(true);
        else
            OutMgr.setLogAppend(false);
    }

    if (tagline.startsWith("OCD1"))
    {
        val = tagline.substring(tagline.indexOf("=")+1).trim();
        if (val.equalsIgnoreCase("false") )
            Config.setOCD1(false);
        else
            Config.setOCD1(true);
    }

    if (tagline.startsWith("Continuous"))
    {
        val = tagline.substring(tagline.indexOf("=")+1).trim();
        if (val.equalsIgnoreCase("true") )
            Config.setContinuous(true);
        else
            Config.setContinuous(false);
    }

    if (tagline.startsWith("GenInterval"))
    {
        val = tagline.substring(tagline.indexOf("=")+1).trim();
        Integer tmpint = new Integer(val);
        setGenInterval(tmpint.intValue());
    }

    if (tagline.startsWith("Compression"))
    {
        val = tagline.substring(tagline.indexOf("=")+1).trim();
        if (val.equalsIgnoreCase("true") )
            Config.setGZip(true);
        else
            Config.setGZip(false);
    }

    if (tagline.startsWith("HTMLExtensions"))
    {
        Config.setExtensions(val);
    }

    if (tagline.startsWith("[FileList]"))
    {
        int i=0;
        tagline = din.readLine();
        String[] tmpList = new String[100];
        while (tagline != null)
        {
            tmpList[i++] = tagline;
            tagline = din.readLine();
        }
        _fileList = new String[i];
        for (int z=0; z<i; z++)
            _fileList[z] = tmpList[z];
    }

```

```

        tagline = din.readLine();
    }

    fin.close();
}
catch (IOException e)
{
    System.out.println("**** ERROR READING CONFIGURATION FILE " + fi );
    return false;
}

_modified=false;

//saveAs("default.cfg");      //saves in default run directory

return true;
}

/***** SAVE TO SPECIFIC FILE *****/
public static boolean saveAs(String fn)
{
    String asciiContents = "";
    File of = new File( fn);

    _configFName = fn;

    try
    {
        FileOutputStream fout = new FileOutputStream(of);
        asciiContents = "[WWWhoosh11Parameters]\r\n" +
            "DateLastRun= " + _dateStamp + "\r\n" +
            "NonameFile= " + _defaultFile + "\r\n" +
            "HTMLExtensions= " + _extensionList + "\r\n" +
            "LastConfigFile= " + _configFName + "\r\n" +
            "TTOUT= " + OutMgr.TTOUT + "\r\n" +
            "OCD1= " + Config.getOCD1() + "\r\n" +

            "Compression= " + Config.getGZip() + "\r\n" +
            "FILEOUT= " + OutMgr.fileOut() + "\r\n" +
            "LOGAPPEND= " + OutMgr.LOG_APPEND + "\r\n" +
            "Domain= " + _domain + "\r\n" +
            "LogFile= " + _logfile + "\r\n" +
            "RootDir= " + _rootDir + "\r\n" +
            "Continuous= " + _continuous + "\r\n" +
            "GenInterval= " + _geninterval + "\r\n" +
            "[FileList]= list\r\n";

        if ( _fileList.length > 0)
            for (int i=0; i< _fileList.length; i++)
            {
                String tmp = _fileList[i];
                asciiContents += tmp + " \n";
            }

        fout.write( asciiContents.getBytes() );
        fout.close();
    }
    catch (IOException e)
    {
        System.out.println("**** ERROR CREATING CONFIG FILE "
            + of.getAbsolutePath() );
        return false;
    }

    return true;
}

```

```
public static boolean save()
{
    String tmp = _rootDir;
    saveAs(_configFName);
    saveAs("WWWwhoosh.INI");
    _rootDir=tmp;
    return true;
}

public static void setMaxDepth(int d)
{ _maxDepth=d;}
public static int getMaxDepth()
{ return _maxDepth;}
public static void setCurDir(String cd)
{ _curDir=cd;}
public static String getCurDir()
{ return _curDir;}
public static void setCurFile(String cf)
{ _curFile=cf;}
public static String getCurFile()
{ return _curFile;}

//{{DECLARE CONTROLS
Config config1 = new Config();
//}}
}
```

```

import java.io.*;
import java.util.*;

/**
 * Reads a configuration file that consists of headers and entries.
 * Headers are of the format <br>
 * <code>[HeaderName]</code><br>
 * entries are of the format <br>
 * <code>EntryName=EntryValue</code>.<p>
 * Example:<br>
 * <code>[Main]<br>
 * HomeDir=/root<br>
 * Timeout=30<br>
 * </code>
 */
public class ConfigFile
{
    private String sFilename;
    private Hashtable sectionTable = null;

    /**
     * Initializes the ConfigFile object from a file.
     *
     * @param _sFilename The path to the file.
     * @exception HmrException if an error occurs opening the file or if
     * the file is of an invalid format.
     */
    public ConfigFile(String _sFilename)
    {
        sFilename = _sFilename;
        try {
            loadFile();
        }
        catch (IOException e)
        {
            OutMgr.print(1,
                " ***** ERROR: could not open file " + sFilename );
        }
        return;
    }

    /**
     * Get a configuration parameter.
     *
     * @param section name of section of the config file
     * @param name name of parameter in the section
     * @return the parameter as a <code>String</code>
     */
    public String get(Object section, Object name) throws HmrValueNotFoundException
    {
        Object valTable = sectionTable.get(section);
        if (valTable == null)
        {
            throw new HmrValueNotFoundException("No section <" + section
                + "> in file <" + sFilename + ">");
        }
        Object objRet = ( (Hashtable)valTable ).get(name);
        if (objRet == null)
        {
            throw new HmrValueNotFoundException("No value for <" + section
                + ">:<" + name + "> in file <" + sFilename + ">");
        }
        return ( (String)objRet );
    }

    /**
     * Get a section of configuration parameters.

```

```

*
* @param    section    name of section of the config file
* @return    a <code>Vector</code> of <code>NameValuePair</code>
*            containing all the parameters in the section
* @see      COM.reitshamer.util.NameValuePair
*/
public Vector getSection(Object section)
{
    Vector v = new Vector();
    Hashtable valTable = (Hashtable)sectionTable.get(section);
    // Loop through section to get values.
    Enumeration eKeys = valTable.keys();
    while (eKeys.hasMoreElements())
    {
        Object elem = eKeys.nextElement();
        NameValuePair nvp = new NameValuePair((String)elem,
            (String)valTable.get(elem));
        v.addElement(nvp);
    }
    return v;
}

/**
 * Re-initialize from the configuration file.
 */
* @exception    COM.reitshamer.error.HmrException if an error occurs
*               opening the file or the file is of an invalid format.
*/
public void reload() throws HmrException
{
    try
    {
        loadFile();
    }
    catch(IOException e)
    {
        throw new HmrException("unable to open file: " + e.getMessage());
    }
}

private void loadFile() throws IOException
{
    int iRet;
    String sLine;

    //Debug.print("opening file <" + sFilename + ">");
    FileReader fread = new FileReader(sFilename);
    BufferedReader bread = new BufferedReader(fread);

    // Create Hashtable which contains Hashtables.
    sectionTable = new Hashtable();
    // Create Hashtable pointer which will point to the current section.
    Hashtable newTable = null;
    String sName;
    int index = 0;

    while (true)
    {
        // Read a line.
        sLine = bread.readLine();
        if (sLine == null)
        {
            break;
        }
        sLine = sLine.trim();
        if ( (sLine.length() == 0) || sLine.substring(0,1).equals(";") )
        {
            continue;
        }
    }
}

```

```
if (sLine.charAt(0) == '[')
{
    // It's a new section name. Reset the 'current section'
    // pointer.
    if (sLine.charAt(sLine.length()-1) == ']')
    {
        sName = sLine.substring(1, sLine.length() - 1);
    }
    else
    {
        sName = sLine.substring(1, sLine.length());
    }
    //Debug.print("got section name <" + sName + ">");
    newTable = new Hashtable();
    sectionTable.put(sName, newTable);
    continue;
}
if (newTable == null)
{
    // There's been no section header. The current section has a
    // label of "".
    newTable = new Hashtable();
    sectionTable.put(new String(""), newTable);
}
StringTokenizer tkzr = new StringTokenizer(sLine, "=");
String name = tkzr.nextToken();
String value = null;
if (!tkzr.hasMoreTokens())
{
    Debug.print("no value for token <" + name + ">");
}
else
{
    value = tkzr.nextToken();
    value = value.trim();
    newTable.put(name, value);
}
//Debug.print("name=<" + name + "> value=<" + value + ">");
}
fread.close();
}
```

```

import java.awt.*;
import java.awt.event.*;
import java.io.File;
import symantec.itools.awt.BorderPanel;

public class ConfirmDialog extends Dialog
{
    public ConfirmDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        //Keep a local reference to the invoking frame
        frame = parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(246,113);
        setVisible(false);
        yesButton.setLabel(" Yes ");
        add(yesButton);
        yesButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        yesButton.setBounds(107,87,64,20);
        noButton.setLabel(" No ");
        add(noButton);
        noButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        noButton.setBounds(170,87,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(4,2,239,88);
        label1.setText("Remove all generated files?");
        label1.setAlignment(java.awt.Label.CENTER);
        borderPanell.add(label1);
        label1.setBounds(21,26,180,23);
        setTitle("WWWhoosh Confirm");
        //}}}

        //{{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        noButton.addActionListener(lSymAction);
        yesButton.addActionListener(lSymAction);

        SymMouseMotion aSymMouseMotion = new SymMouseMotion();
        this.addMouseMotionListener(aSymMouseMotion);
        //}}}
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;
    }
}

```

```

        // Adjust components according to the insets
        setSize(getInsets().left + getInsets().right + d.width, getInsets().top +
getInsets().bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(getInsets().left, getInsets().top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

    public ConfirmDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    /**
     * Shows or hides the component depending on the boolean flag b.
     * @param b if true, show the component; otherwise, hide the component.
     * @see java.awt.Component#isVisible
     */
    public void setVisible(boolean b)
    {
        if(b)
        {
            Rectangle bounds = getParent().getBounds();
            Rectangle abounds = getBounds();

            setLocation(bounds.x + (bounds.width - abounds.width) / 2,
                bounds.y + (bounds.height - abounds.height) / 2);
            Toolkit.getDefaultToolkit().beep();
        }
        super.setVisible(b);
    }

    boolean _deleteOK = false;

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;
    // Invoking frame
    Frame frame = null;

    /**{DECLARE_CONTROLS
    java.awt.Button yesButton = new java.awt.Button();
    java.awt.Button noButton = new java.awt.Button();
    symantec.itools.awt.BorderPanel borderPanel1 = new symantec.itools.awt.BorderPanel();
    java.awt.Label label1 = new java.awt.Label();
    /**}

    class SymAction implements java.awt.event.ActionListener
    {
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
            Object object = event.getSource();
            if (object == yesButton)
                yesButton_ActionPerformed(event);
            else if (object == noButton)
                noButton_ActionPerformed(event);
        }
    }

    boolean deleteOK()
    {return _deleteOK;}

    void yesButton_ActionPerformed(java.awt.event.ActionEvent event)
    {

```



```

        _deleteOK = true;
        try {
            this.dispose();
        } catch (Exception e) {
        }
    }

    void noButton_ActionPerformed(java.awt.event.ActionEvent event)
    {
        // to do: code goes here.

        noButton_ActionPerformed_Interaction1(event);
    }

    void noButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
    {
        try {
            this.dispose();
        } catch (Exception e) {
        }
    }

    class SymWindow extends java.awt.event.WindowAdapter
    {
        public void windowClosed(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == ConfirmDialog.this)
                ConfirmDialog_WindowClosed(event);
        }

        public void windowClosing(java.awt.event.WindowEvent event)
        {
            Object object = event.getSource();
            if (object == ConfirmDialog.this)
                ConfirmDialog_WindowClosing(event);
        }
    }

    void ConfirmDialog_WindowClosing(java.awt.event.WindowEvent event)
    {
        // to do: code goes here.

        ConfirmDialog_WindowClosing_Interaction1(event);
    }

    void ConfirmDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
    {
        try {
            this.dispose();
        } catch (Exception e) {
        }
    }

    class SymMouseMotion extends java.awt.event.MouseMotionAdapter
    {
        public void mouseMoved(java.awt.event.MouseEvent event)
        {
            Object object = event.getSource();
            if (object == ConfirmDialog.this)
                ConfirmDialog_MouseMoved(event);
        }
    }

```

```
}  
void ConfirmDialog_MouseMoved(java.awt.event.MouseEvent event)  
{  
    // to do: code goes here.  
}  
void ConfirmDialog_WindowClosed(java.awt.event.WindowEvent event)  
{  
    // to do: code goes here.  
}  
}
```

```

import java.awt.*;
import java.awt.event.*;
import java.io.File;
import symantec.itools.awt.BorderPanel;

public class ConfirmSaveExitDialog extends Dialog
{
    public ConfirmSaveExitDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        //Keep a local reference to the invoking frame
        frame = parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(300,107);
        setVisible(false);
        yesButton.setLabel(" Yes ");
        add(yesButton);
        yesButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        yesButton.setBounds(156,80,64,20);
        noButton.setLabel(" No ");
        add(noButton);
        noButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        noButton.setBounds(221,80,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(7,8,290,71);
        labell.setText("Save changes to configuration file?");
        labell.setAlignment(java.awt.Label.CENTER);
        borderPanell.add(labell);
        labell.setBounds(31,16,204,23);
        setTitle("WWWhoosh - Save Config");
        //}}

        //{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        noButton.addActionListener(lSymAction);
        yesButton.addActionListener(lSymAction);

        SymMouseMotion aSymMouseMotion = new SymMouseMotion();
        this.addMouseMotionListener(aSymMouseMotion);
        //}}
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;
    }
}

```

```

        // Adjust components according to the insets
        setSize(getInsets().left + getInsets().right + d.width, getInsets().top +
getInsets().bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(getInsets().left, getInsets().top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

    public ConfirmSaveExitDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    /**
     * Shows or hides the component depending on the boolean flag b.
     * @param b if true, show the component; otherwise, hide the component.
     * @see java.awt.Component#isVisible
     */
    public void setVisible(boolean b)
    {
        if(b)
        {
            Rectangle bounds = getParent().getBounds();
            Rectangle abounds = getBounds();

            setLocation(bounds.x + (bounds.width - abounds.width) / 2,
                bounds.y + (bounds.height - abounds.height) / 2);
            Toolkit.getDefaultToolkit().beep();
        }
        super.setVisible(b);
    }

    boolean _deleteOK = false;

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;
    // Invoking frame
    Frame frame = null;

    /**{DECLARE_CONTROLS
    java.awt.Button yesButton = new java.awt.Button();
    java.awt.Button noButton = new java.awt.Button();
    symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
    java.awt.Label labell = new java.awt.Label();
    //}}

    class SymAction implements java.awt.event.ActionListener
    {
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
            Object object = event.getSource();
            if (object == yesButton)
                yesButton_ActionPerformed(event);
            else if (object == noButton)
                noButton_ActionPerformed(event);
        }
    }

    boolean deleteOK()
    {return _deleteOK;}

    void yesButton_ActionPerformed(java.awt.event.ActionEvent event)
    {

```

```
Config.save();

    try {
        this.dispose();
    } catch (Exception e) {
    }
}

void noButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    noButton_ActionPerformed_Interaction1(event);
}

void noButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosed(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == ConfirmSaveExitDialog.this)
            ConfirmSaveExitDialog_WindowClosed(event);
    }

    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == ConfirmSaveExitDialog.this)
            ConfirmSaveExitDialog_WindowClosing(event);
    }
}

void ConfirmSaveExitDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    ConfirmSaveExitDialog_WindowClosing_Interaction1(event);
}

void ConfirmSaveExitDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymMouseMotion extends java.awt.event.MouseMotionAdapter
{
    public void mouseMoved(java.awt.event.MouseEvent event)
    {
        Object object = event.getSource();
        if (object == ConfirmSaveExitDialog.this)
            ConfirmSaveExitDialog_MouseMoved(event);
    }
}
```

```
    }  
}  
  
void ConfirmSaveExitDialog_MouseMoved(java.awt.event.MouseEvent event)  
{  
    // to do: code goes here.  
}  
  
void ConfirmSaveExitDialog_WindowClosed(java.awt.event.WindowEvent event)  
{  
    // to do: code goes here.  
}  
}
```

```

import java.io.*;
import java.util.Date;
import java.util.zip.GZIPOutputStream;
import java.util.zip.GZIPInputStream;

import WWWhooshCommon.*;

/*
This class is responsible for knowing the details of the OC file formats
and hiding those details from the Optimizer.

It handles writing specific subsections of the format to a file.
*/

public class FileFormatMgr
{
    String _filename=null;
    String _filedir=null;
    File _filehandle=null;
    FileOutputStream _filestream=null;
    DataOutputStream _dos;

    //instantiate with file name and directory of input or output file
    public FileFormatMgr(String filedir, String filename)
    {
        _filename=filename;
        _filedir=filedir;
        _filehandle = new File(filedir, filename);
        try{
            _filestream = new FileOutputStream(_filehandle);
            _dos = new DataOutputStream(_filestream);
        }
        catch (IOException ioe)
        {
            OutMgr.print(" **** ERROR: failed to access output file for writing ****" +
                _filename);
        }
        //{{{INIT_CONTROLS
        //}}}
    }

    //use this constructor while we still have code that creates the data streams
    //in Optimizer. Eventually move all that code into this class so it's
    //the only one actually touching files
    public FileFormatMgr()
    {}

    private void readHeader(BufferedInputStream bis)
    {
    }

    //write header using filestream member from instantiation
    private void writeHeader()
    {
        WhooshHeader whooshHDR = new WhooshHeader();
        whooshHDR.writeTo(_filestream);
    }

    public void writeHeader(FileOutputStream fs)
    {
        WhooshHeader whooshHDR = new WhooshHeader();
        whooshHDR.writeTo(fs);
    }

    public String readCoherency(BufferedInputStream bis) throws IOException
    {
    }

```

```

//TODO: this code should be common with that in OCResponse
int iRet=0;

WhooshHeader whoosh= new WhooshHeader();
byte[] fixbuf = new byte[1028];

try
{
    whoosh.readFrom(bis); //TODO: capture length for stats
    iRet = bis.read(fixbuf);
}
catch(IOException e)
{
}

ByteArrayOutputStream baos = new ByteArrayOutputStream();

boolean gzipformat=false;
if (fixbuf[0] == 0x1F ) //next byte should be 0x8b == 139
    gzipformat = true;

String retstr=null;

try
{
    while (iRet != -1)
    {
        baos.write(fixbuf, 0, iRet);
        iRet = bis.read(fixbuf);
    }

    byte[] ba = baos.toByteArray(); //get variable-length as byte array

    if (gzipformat)
    {
        //all this recasting seems a bit silly but required by type collection
        ByteArrayInputStream bais = new ByteArrayInputStream(ba); //construct stream around
        GZIPInputStream gzin = new GZIPInputStream(bais); //construct gzip array from

        //now we're ready to uncompress stream into another byte array
        ByteArrayOutputStream uncompressedstream = new ByteArrayOutputStream();
        int fullcount = gzin.read(fixbuf);
        while (fullcount != -1)
        {
            uncompressedstream.write(fixbuf,0,fullcount);
            fullcount = gzin.read(fixbuf);
        }
        ba = uncompressedstream.toByteArray();
    }

    retstr = new String(ba);
}
catch(IOException e)
{
}

return retstr;
}

//Write entire Coherency file, given the info in string form.
//Also write a _noname file, if this is a default HTML file.
public void writeCoherencyFile(String coherency)
{
    //start by writing the header
    writeHeader();

    try { //TODO: put compression into its own utility method

```



```

if (coherency != null)
{
    if (Config.getGZip())
    {
        GZIPOutputStream gzout = new GZIPOutputStream(_dos);    //careful-writes 10-byte
header!
        gzout.write(coherency.getBytes());
        gzout.close();
    }
    else
    {
        _dos.write(coherency.getBytes());
    }
}

_dos.close();

Stats._bytesWritten+=_filehandle.length();
}
catch (IOException e)
{
    OutMgr.print(" **** ERROR writing to output file **** " + _filename );
}

//File noname - copy generated OC file to oc_noname.htm
/*
File noname = null;
if (fn.equalsIgnoreCase( Config.getDefaultFile() ) )
    noname = new File( fd, "oc_noname.htm" );
if (noname !=null)
{
    try
    {
        FileOutputStream xnonameout = new FileOutputStream(noname);
        if (OCdata != null)
        {
            if (Config.getGZip())
            {
                GZIPOutputStream nonameout = new GZIPOutputStream(xnonameout);
                nonameout.write(OCdata.getBytes() );

                nonameout.close();
            }
            else
            {
                xnonameout.write(OCdata.getBytes() );
                xnonameout.close();
            }
        }

        Stats._numOutFiles++;
        xnonameout.close();
        Stats._bytesWritten+=noname.length();
    }
    catch (IOException e)
    {
        OutMgr.print(" **** ERROR: Exception writing coherency data **** " +
ofile );
    }
}

*/

OutMgr.print(OutMgr.INCSTATUS, "      Created: " + OutMgr.chopPath(_filename)
+ "      size: " + _filehandle.length() + " bytes " );

/*
if (noname != null)
    OutMgr.print(OutMgr.INCSTATUS, "      Created: oc_noname.htm" );
*/

```

}

```
//({DECLARE_CONTROLS  
//})
```

}

```

import java.awt.*;
import java.io.File;
import symantec.itools.awt.ImagePanel;
import symantec.itools.awt.BorderPanel;
import com.sun.java.swing.JProgressBar;

public class InputsDialog extends Dialog {

    public InputsDialog(WWWwhooshServer parent, boolean modal)
    {
        super(parent, modal);
        _parentapp=parent;
        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(484,403);
        setVisible(false);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        okButton.setBounds(280,374,64,20);
        button2.setLabel("Cancel");
        add(button2);
        button2.setFont(new Font("Dialog", Font.PLAIN, 12));
        button2.setBounds(344,374,64,20);
        button1.setLabel("Apply");
        add(button1);
        button1.setFont(new Font("Dialog", Font.PLAIN, 12));
        button1.setBounds(406,374,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setLabel("  Server Setup ");
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(10,12,464,170);
        label7.setText("text");
        borderPanell.add(label7);
        label7.setBounds(134,222,100,40);
        label11.setText("Domain Name:");
        borderPanell.add(label11);
        label11.setBounds(8,8,129,19);
        label13.setText("Default Page:");
        borderPanell.add(label13);
        label13.setBounds(8,80,129,19);
        textField4.setText("htm html shtml");
        borderPanell.add(textField4);
        textField4.setBounds(139,32,300,20);
        textField1.setText("www.yourdomainhere.com");
        borderPanell.add(textField1);
        textField1.setBounds(139,8,300,20);
        label4.setText("HTML File Extensions:");
        borderPanell.add(label4);
        label4.setFont(new Font("Dialog", Font.PLAIN, 12));
        label4.setBounds(8,32,129,19);
    }
}

```

```

subobjextField.setText("gif jpg jpeg");
borderPanel1.add(subobjextField);
subobjextField.setBounds(139,56,300,20);
label6.setText("Subobject Extensions:");
borderPanel1.add(label6);
label6.setBounds(8,56,129,19);
textField3.setText("index.html");
borderPanel1.add(textField3);
textField3.setBounds(139,80,300,20);
domainRootField.setText("domain directory");
borderPanel1.add(domainRootField);
domainRootField.setBounds(139,104,233,20);
label9.setText("Domain Root:");
borderPanel1.add(label9);
label9.setBounds(8,104,129,19);
rootButton.setLabel("Browse...");
borderPanel1.add(rootButton);
rootButton.setFont(new Font("Dialog", Font.PLAIN, 12));
rootButton.setBounds(376,104,64,20);
try {
    borderPanel2.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setLabel(" Inputs for Processing ");
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel2.setLayout(null);
add(borderPanel2);
borderPanel2.setBounds(10,191,465,182);
borderPanel2.add(list2);
list2.setBounds(9,47,366,88);
borderPanel2.add(textField2);
textField2.setBounds(63,6,309,20);
insertButton.setLabel("Insert File");
borderPanel2.add(insertButton);
insertButton.setFont(new Font("Dialog", Font.PLAIN, 12));
insertButton.setBounds(378,55,64,20);
deleteButton.setLabel("Delete");
borderPanel2.add(deleteButton);
deleteButton.setFont(new Font("Dialog", Font.PLAIN, 12));
deleteButton.setBounds(378,106,64,20);
browseButton.setLabel("Browse...");
borderPanel2.add(browseButton);
browseButton.setFont(new Font("Dialog", Font.PLAIN, 12));
browseButton.setBounds(376,6,64,20);
insPathButt.setLabel("Insert Path");
borderPanel2.add(insPathButt);
insPathButt.setFont(new Font("Dialog", Font.PLAIN, 12));
insPathButt.setBounds(378,80,64,20);
label5.setText("Add New:");
borderPanel2.add(label5);
label5.setBounds(8,5,52,23);
try {
    borderPanel3.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel3.setPaddingRight(1);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel3.setPaddingBottom(1);
}
catch(java.beans.PropertyVetoException e) { }

```

```

    try {
        borderPanel3.setPaddingTop(1);
    }
    catch(java.beans.PropertyVetoException e) { }
    try {
        borderPanel3.setPaddingLeft(1);
    }
    catch(java.beans.PropertyVetoException e) { }
    borderPanel3.setLayout(null);
    borderPanel2.add(borderPanel3);
    borderPanel3.setBounds(10,33,432,5);
    setTitle("WWWhoosh Inputs: Files to Process");
    //}}

    //{{REGISTER LISTENERS
    SymWindow aSymWindow = new SymWindow();
    this.addWindowListener(aSymWindow);
    SymAction lSymAction = new SymAction();
    okButton.addActionListener(lSymAction);
    button2.addActionListener(lSymAction);
    insertButton.addActionListener(lSymAction);
    deleteButton.addActionListener(lSymAction);
    browseButton.addActionListener(lSymAction);
    list2.addActionListener(lSymAction);
    insPathButt.addActionListener(lSymAction);
    button1.addActionListener(lSymAction);
    textField2.addActionListener(lSymAction);
    rootButton.addActionListener(lSymAction);
    //}}
}

public InputsDialog(WWWhooshServer parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();
    }
}

```

```

        setLocation(bounds.x + (bounds.width - abounds.width)/ 2,
                    bounds.y + (bounds.height - abounds.height)/2);
    }

    super.setVisible(b);
}

WWWhooshServer _parentapp=null;
//{{{DECLARE_CONTROLS
java.awt.Button okButton = new java.awt.Button();
java.awt.Button button2 = new java.awt.Button();
java.awt.Button button1 = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanel1 = new symantec.itools.awt.BorderPanel();
java.awt.Label label7 = new java.awt.Label();
java.awt.Label label11 = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
java.awt.TextField textField4 = new java.awt.TextField();
java.awt.TextField textField1 = new java.awt.TextField();
java.awt.Label label4 = new java.awt.Label();
java.awt.TextField subobjextField = new java.awt.TextField();
java.awt.Label label6 = new java.awt.Label();
java.awt.TextField textField3 = new java.awt.TextField();
java.awt.TextField domainRootField = new java.awt.TextField();
java.awt.Label label9 = new java.awt.Label();
java.awt.Button rootButton = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanel2 = new symantec.itools.awt.BorderPanel();
java.awt.List list2 = new java.awt.List(4);
java.awt.TextField textField2 = new java.awt.TextField();
java.awt.Button insertButton = new java.awt.Button();
java.awt.Button deleteButton = new java.awt.Button();
java.awt.Button browseButton = new java.awt.Button();
java.awt.Button insPathButt = new java.awt.Button();
java.awt.Label label5 = new java.awt.Label();
symantec.itools.awt.BorderPanel borderPanel3 = new symantec.itools.awt.BorderPanel();
//}}}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
        else if (object == button2)
            button2_ActionPerformed(event);
        else if (object == insertButton)
            insertButton_ActionPerformed(event);
        else if (object == deleteButton)
            deleteButton_ActionPerformed(event);
        else if (object == browseButton)
            browseButton_ActionPerformed(event);
        else if (object == list2)
            list2_ActionPerformed(event);
        else if (object == insPathButt)
            insPathButt_ActionPerformed(event);
        else if (object == button1)
            button1_ActionPerformed(event);
        else if (object == textField2)
            textField2_ActionPerformed(event);
        else if (object == rootButton)
            rootButton_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{

```

```

        okButton_ActionPerformed_Interaction1(event);
    }

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    Config.setExtensions(textField4.getText());
    Config.setDefaultFile(textField3.getText() );
    Config.setDomain( textField1.getText() );
    String[] flist = list2.getItems();
    Config.setRootDir( domainRootField.getText() );

    //warn them about bad filenames, but allow them to proceed
    for (int i =0; i< flist.length;i++)
    {
        File testfile = new File(flist[i]);
        if (!testfile.exists())
            OutMgr.print("*** WARNING - File Not Found *** "
                + testfile.getAbsolutePath() );
    }

    Config.setFiles(flist );
    _parentapp.updateStatLabels();

    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowOpened(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == InputsDialog.this)
            InputsDialog_WindowOpened(event);
    }

    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == InputsDialog.this)
            InputsDialog_WindowClosing(event);
    }
}

void InputsDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.
    InputsDialog_WindowClosing_Interaction1(event);
}

void InputsDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    ConfirmSaveDialog csd = null;

    try {
        // ConfirmSaveDialog Create and show as modal
        csd = new ConfirmSaveDialog( _parentapp, true);
        csd.setVisible(true);
    }
    catch (Exception e) { }

    if (csd.SAVEIT)
        updateConfig();
}

```

```

setVisible(false);
}
void button2_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    button2_ActionPerformed_Interaction1(event);
}

void button2_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    _parentapp.updateStatLabels();
    try {
        // InputsDialog Hide the InputsDialog
        this.setVisible(false);
    } catch (Exception e) {
    }
}

void InputsDialog_WindowOpened(java.awt.event.WindowEvent event)
{
    // to do: code goes here.
    textField4.setText(Config.getExtensions() );
    textField3.setText(Config.getDefaultFile());
    textField1.setText(Config.getDomain() );
    domainRootField.setText(Config.getRootDir());
    textField2.setText(Config.getRootDir());
    String[] flist = Config.GetFiles();

    if (flist !=null)
        for (int i=0; i<flist.length;i++)
            list2.add(flist[i]);
}

void insertButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    String fspec = textField2.getText();
    if ((fspec != null) && !fspec.equals(""))
        list2.add(fspec);
}

void deleteButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    String item = list2.getSelectedItem();
    if (item != null)
        list2.remove(item);
}

void browseButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    FileDialog fileDialog1 = new java.awt.FileDialog((Frame)this.getParent());

    textField2.setText(domainRootField.getText());
    if (textField2.getText() != "")
        fileDialog1.setDirectory( textField2.getText() );
    else
        fileDialog1.setDirectory( Config.getRootDir() );

    fileDialog1.setFile(Config.getDefaultFile() );
    fileDialog1.setVisible(true);
    String inFileName = fileDialog1.getFile();
    String dirname = fileDialog1.getDirectory();
    if (inFileName !=null)
        textField2.setText(dirname + inFileName);
    Config.setCurDir(dirname);
    Config.setDomain(textField1.getText());
}

```



```

    }

    void list2_ActionPerformed(java.awt.event.ActionEvent event)
    {
        textField2.setText(list2.getSelectedItem() );
    }

    void insPathButt_ActionPerformed(java.awt.event.ActionEvent event)
    {
        String fspec = textField2.getText();
        if ((fspec != null) && !fspec.equals(""))
        {
            String pathpart = fspec;
            int lastslash = fspec.lastIndexOf("\\");
            if (lastslash > -1)
                pathpart = fspec.substring(0,lastslash);
            list2.add(pathpart);
        }
    }

    void updateConfig()
    {
        Config.setExtensions(textField4.getText());
        Config.setDefaultFile(textField3.getText() );
        Config.setDomain( textField1.getText() );
        Config.setRootDir( domainRootField.getText());
        String[] flist = list2.getItems();
        Config.setFiles(flist );
        _parentapp.updateStatLabels();
    }

    void button1_ActionPerformed(java.awt.event.ActionEvent event)
    {
        updateConfig();
    }

    void textField2_ActionPerformed(java.awt.event.ActionEvent event)
    {
        insPathButt_ActionPerformed(event);
    }

    void rootButton_ActionPerformed(java.awt.event.ActionEvent event)
    {
        FileDialog fileDialog2 = new java.awt.FileDialog((Frame)this.getParent());

        if (domainRootField.getText() != "")
            fileDialog2.setDirectory( textField2.getText() );
        else
            fileDialog2.setDirectory( Config.getRootDir() );

        //fileDialog2.setFile(Config.getDefaultFile() );
        fileDialog2.setVisible(true);

        String inFileName = fileDialog2.getFile();
        String dirname = fileDialog2.getDirectory();
        Config.setRootDir(dirname);

        if (inFileName !=null)
        {
            domainRootField.setText( Config.getRootDir() );
            textField2.setText( Config.getRootDir() );
        }
        Config.setCurDir(dirname);
    }

```

```
        Config.setRootDir(domainRootField.getText());  
    }  
}
```

```

/*
 * A basic extension of the java.awt.Dialog class
 */

import java.awt.*;
import symantec.itools.awt.BorderPanel;

public class LogLevelDialog extends Dialog
{
    Frame _parent;

    public LogLevelDialog(Frame parent)
    {
        super(parent);
        _parent=parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(320,234);
        setVisible(false);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setBounds(114,204,64,20);
        button1.setLabel("Cancel");
        add(button1);
        button1.setBounds(179,204,64,20);
        button2.setLabel("Apply");
        add(button2);
        button2.setBounds(244,204,64,20);
        try {
            borderPanel3.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel3.setLabel(" Logging ");
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel3.setPaddingRight(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel3.setPaddingBottom(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel3.setPaddingTop(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel3.setPaddingLeft(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanel3.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanel3.setLayout(null);
        add(borderPanel3);
        borderPanel3.setBounds(10,11,299,188);
        checkbox4.setLabel("Print Incremental Status ");
    }
}

```

```

borderPanel3.add(checkbox4);
checkbox4.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox4.setBounds(16,78,179,26);
checkbox1.setLabel("Print Filenames");
borderPanel3.add(checkbox1);
checkbox1.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox1.setBounds(16,6,116,24);
checkbox2.setLabel("Print Full Paths");
borderPanel3.add(checkbox2);
checkbox2.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox2.setBounds(16,30,116,24);
checkbox3.setLabel("Print Debug Messages");
borderPanel3.add(checkbox3);
checkbox3.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox3.setBounds(16,53,179,26);
checkbox5.setLabel("Append to Logfile");
borderPanel3.add(checkbox5);
checkbox5.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox5.setBounds(16,104,179,26);
borderPanel3.add(textField1);
textField1.setBounds(57,137,227,20);
label1.setText("Logfile");
borderPanel3.add(label1);
label1.setBounds(16,136,45,22);
//))

//((REGISTER_LISTENERS
SymWindow aSymWindow = new SymWindow();
this.addWindowListener(aSymWindow);
SymAction lSymAction = new SymAction();
button2.addActionListener(lSymAction);
okButton.addActionListener(lSymAction);
button1.addActionListener(lSymAction);
//))
}

public LogLevelDialog(Frame parent, boolean modal)
{
    this(parent);
    setModal(modal);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }
    fComponentsAdjusted = true;
}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

```

```

public LogLevelDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

/**
 * Shows or hides the component depending on the boolean flag b.
 * @param b if true, show the component; otherwise, hide the component.
 * @see java.awt.Component#isVisible
 */
public void setVisible(boolean b)
{
    if(b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }
    super.setVisible(b);
}

//{{{DECLARE_CONTROLS
java.awt.Button okButton = new java.awt.Button();
java.awt.Button button1 = new java.awt.Button();
java.awt.Button button2 = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanel3 = new symantec.itools.awt.BorderPanel();
java.awt.Checkbox checkbox4 = new java.awt.Checkbox();
java.awt.Checkbox checkbox1 = new java.awt.Checkbox();
java.awt.Checkbox checkbox2 = new java.awt.Checkbox();
java.awt.Checkbox checkbox3 = new java.awt.Checkbox();
java.awt.Checkbox checkbox5 = new java.awt.Checkbox();
java.awt.TextField textField1 = new java.awt.TextField();
java.awt.Label label1 = new java.awt.Label();
//}}}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowOpened(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == LogLevelDialog.this)
            LogLevelDialog_WindowOpened(event);
    }

    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == LogLevelDialog.this)
            LogLevelDialog_WindowClosing(event);
    }
}

void LogLevelDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    ConfirmSaveDialog csd = null;

    try {
        // ConfirmSaveDialog Create and show as modal
        csd = new ConfirmSaveDialog(_parent, true);
        csd.setVisible(true);
    }
    catch (Exception e) { }

    if (csd.SAVEIT)
    {
        OutMgr.PRINT_FILENAMES =checkboxox1.getState();
    }
}

```

```

        OutMgr.PRINT_FULLNAMES =checkbox2.getState();
        OutMgr.PRINT_DEBUGMSG  =checkbox3.getState();
        OutMgr.PRINT_INCSTATUS =checkbox4.getState();
        OutMgr.LOG_APPEND      =checkbox5.getState();
        Config.setLogFile(textField1.getText());
        OutMgr.setLogfile(textField1.getText());
    }
    setVisible(false);
}

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == button2)
            button2_ActionPerformed(event);
        else if (object == okButton)
            okButton_ActionPerformed(event);
        else if (object == button1)
            button1_ActionPerformed(event);
    }
}

void button2_ActionPerformed(java.awt.event.ActionEvent event)
{
    OutMgr.PRINT_FILENAMES =checkbox1.getState();
    OutMgr.PRINT_FULLNAMES =checkbox2.getState();
    OutMgr.PRINT_DEBUGMSG  =checkbox3.getState();
    OutMgr.PRINT_INCSTATUS =checkbox4.getState();
    OutMgr.LOG_APPEND = checkbox5.getState();
    Config.setLogFile(textField1.getText());
    OutMgr.setLogfile(textField1.getText());
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    button2_ActionPerformed( event);

    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // LogLevelDialog Hide the LogLevelDialog
        this.setVisible(false);
    } catch (Exception e) {
    }
}

void LogLevelDialog_WindowOpened(java.awt.event.WindowEvent event)
{
    // to do: code goes here.
    checkbox1.setState(OutMgr.PRINT_FILENAMES);
    checkbox2.setState(OutMgr.PRINT_FULLNAMES);
    checkbox3.setState(OutMgr.PRINT_DEBUGMSG);
    checkbox4.setState(OutMgr.PRINT_INCSTATUS );
    checkbox5.setState(OutMgr.LOG_APPEND);
    textField1.setText(Config.getLogFile());
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    hide();
}
}

```



```

import java.util.*;
import java.util.zip.*;
import java.io.*;
import WWWhooshCommon.*;

/***** OPTIMIZER *****/
//
// @author Mark Palmer
//
// @version 0.1 - initial beta release, 1-JUL-1999
// @version 1.0 - support version info in header, compression optional
// @version 2.0 - support incremental updates
//
//
// Modifications:
//
// This class is responsible for taking an input directory or file spec and outputting
// OC and OCD files for it. It interacts with the GUI and Config classes to decide
// how much to do. The Optimizer is a Thread so it can be stopped/started by GUI
// Button callbacks running in the main thread.
//
// BUILD      DATE      COMMENTS
//
// 746        4-Aug-99
//
//          1- buttons reworked to be better placed and uniform sizes
//
//          2- error messages made more uniform (case and severity levels)
//
//          3- STOP button message changed -
//             changed the message so it says "Finishing current input" as
//             soon as you hit the button, then "Generation Halted" is printed
//             after the current input is done processing.
//
//          4- changed domain root specification to use browse field
//
//          13-Aug-99   More GUI refinement:
//                      made some dialogs modal
//                      moved buttons off panels
//                      set -noresize
//
//          14-Aug-99   performance issues - initialization time
//
//          18-Aug-99   rework logging, add -append
//
//          19-Aug-99   create/integrate FileFormatMgr class to write header
//                      info before OC and OCD info
//
//          30-Aug-99   add dialog to save changes on input dialog close, fix logging bug
//                      consolidate WhooshHeader class - common across client and server
//
//          1-Sep-99    move
//                      Incremental updates: generate OC BEFORE OCD, and compare all
//                      timestamps.
//                      if none have changed, skip writing OC and generating OCD
//
//          2-Sep-99    renamed all classes and vars and labels from ICDS to WWWhoosh
//
/***** OPTIMIZER *****/
public class Optimizer extends Thread
{
    Hashtable _OCdupcheck= new Hashtable(); //to detect duplicate entries while building OC
    Hashtable _OCDdupcheck= new Hashtable(); //to detect duplicates while building OCD
    Hashtable notfound= new Hashtable(); //to prevent duplicate "not found" error messages

    static String _linebuf=""; //buffer for reading lines of HTML files
    WWWhooshServer _server=null; //handle to GUI thread for updating statistics

```



```

//legal characters - we hash into this string to check for legal substrings
//TODO: convert so we check whether numeric value is in alphanumeric range
String OKchars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890.-_/\\";
String[] _flist=null; //list of files to be processed
boolean haltmsg=false;
boolean done=false;

//main method called by Thread.start();
public void run()
{
    while (true)
    {
        Stats.clear();
        OutMgr.print("-----> " +
            "Site optimization started at " + new Date().toLocaleString()
            + " <-----" );
        OutMgr.print(" ");

        reset();
        // _flist = Config.GetFiles();

        if (_flist != null)
        {
            for (int i = 0; i < _flist.length; i++)
            {
                File listin=new File(_flist[i]);

                if(!listin.exists())
                {
                    OutMgr.print(" **** ERROR: Specified input file not found **** " +
                        OutMgr.chopPath( OutMgr.chopPath(listin.getAbsolutePath()) ) + ". - skipping"
                    );
                }
                else
                {
                    if (listin.isDirectory() )
                        process(_flist[i].trim(), null);
                    else
                        process(listin.getParent(), listin.getName() );
                }
            }
        }

        OutMgr.print("\r\n");
        Config.setDate(new Date().toLocaleString());

        OutMgr.print("Run Summary Statistics: " + Config.getDomain() );
        OutMgr.print(" " + Stats._numChecks + " unique checks generated");
        OutMgr.print(" " + Stats._notFound + " SRC file missing errors");
        OutMgr.print(" " + Stats._numOutFiles + " files generated");
        OutMgr.print(" " + Stats._numInFiles + " files read");
        OutMgr.print(" " + Stats._bytesRead + " bytes read");
        OutMgr.print(" " + Stats._bytesWritten + " bytes written");

        // TODO: consider reporting these statistics
        // density and efficiency are conceived as simple relative numeric measures of the
        // average round-trips and bandwidth savings to be expected by processing a site using Optimizer
        //
        // OutMgr.print(" " + "Density: " + Stats._numChecks/Stats._numInFiles + "\n");
        // OutMgr.print(" " + "Efficiency: " + Stats._bytesRead/Stats._bytesWritten + "
\n\n");

        OutMgr.print(" ");
        OutMgr.print("-----> " +
            "Site optimization ended at " + new Date().toLocaleString()
            + " <-----" );

        if (Config.getContinuous() == false)

```

```

        return;

    try
    {
        long tmpst = System.currentTimeMillis() + Config.getGenInterval() * 60000;
        Date tmpd = new Date(tmpst);
        OutMgr.print("-----> Next Run Time: " + tmpd.toLocaleString() );

        sleep( Config.getGenInterval() * 60000 );
        catch (InterruptedException ie)
        {}

    } //end of forever

}

public synchronized void finishCycle()
{done=true;
notifyAll();}

public synchronized void waitForCycle()
{
while (!done)
    try{
        this.wait();
    }
    catch (InterruptedException ie)
    {}
}

//Accessor and utility methods
/***** UTILITY METHODS *****/
public void setFileList(String[] flist)
//@input flist - array of file or directory names to be processed
{ _flist = flist; }

//clear duplicate-check caches
public void reset()
{
    _OCDupcheck = new Hashtable();
    _OCDdupcheck = new Hashtable();
}

public Optimizer(WWWhooshServer s)
//@input s - handle to GUI thread
{ _server=s; //((INIT_CONTROLS
//))
}

//heck whether a given file has been generated by this utility
//in order to avoid processing files that we generated
//
boolean isOCFile(String infile)
{
    if ( infile.substring(0,3).equalsIgnoreCase("oc ") ||
        infile.substring(0,4).equalsIgnoreCase("ocd_"))
        return true;
    return false;
}

//general rule for extracting filenames: if valid extension after last dot
/*****READ LEGAL SUBSTRING *****/
public String readLegalString(BufferedReader f)
{
    if (f== null)
        return null;
}

```

```

String result="";

int ext=-1;
while (result == "")
{
    //TODO: handle different length extensions - hack here assumes all <= 4
    //TODO: get the list of extensions from user config
    if (_linebuf != null)
    {
        ext = _linebuf.indexOf(" src");
        if (ext == -1)
            ext = _linebuf.indexOf(".gif");
        if (ext == -1)
            ext = _linebuf.indexOf(".jpg");
        if (ext == -1)
            ext = _linebuf.indexOf(".htm");
    }

    if (ext != -1)                                // extensions in this line
    {
        //we have a subobject extension starting at ext
        //lop off stuff to the right
        char[] toend = _linebuf.toCharArray();
        int endpath= _linebuf.length()-1;
        int startpath=0;

        while ((OKchars.indexOf(toend[startpath]) == -1) && (startpath < endpath))    //skip
            startpath++;

        endpath=startpath;

        while ((endpath < _linebuf.length()) && (OKchars.indexOf(toend[endpath]) != -1)
        )//get good
            endpath++;

        result = _linebuf.substring(startpath,endpath);
        _linebuf = _linebuf.substring(endpath);
    }
    else
        try //no extension found, get another line
        {
            result = "";
            _linebuf=f.readLine();
            if (_linebuf==null)
            {
                f.close();
                return null;
            }

            _linebuf = _linebuf.toLowerCase();
        }
        catch (IOException ioe)
        {
            OutMgr.print("*****ERROR - IO Exception reading input stream *** " + ioe );
            result=null;
        }
    return result;
}

/***** DETECT SUBOBJECT FILENAME *****/
//general rule for detecting subobject filenames: if valid extension after last dot
// TODO: how do we handle filenames with no extension?
//      perhaps by opening each file and looking for the header bytes of a .gif file, etc.
//
public static boolean isSubobject(String f)
//@input f - filename to be checked
//@returns whether filename is considered a subobject for building checks and OCDs
{

```

```

    if (f== null)
        return false;

    String filename = f.toLowerCase();
    int dot = filename.lastIndexOf(".gif");

    dot = filename.lastIndexOf(".");
    if (dot == -1)
        return false;

    //rule out filenames that continue past the extension
    if ( (filename.length() < dot+4) || (filename.length() > dot+5))
        return false;

    String ext = filename.substring(dot+1, dot+4) ;

    if (Config.getSubobjectExtensions().indexOf(ext.toLowerCase()) != -1)
        (return true;)
    else
        return false;
}

/***** utility to unify slash direction, lowercase, and eliminate back refs
*****/
//
// There can be many ways of referring to the same file, and the same file spec gets reported
// differently by various system calls and differently when these same Java calls are run on
// different OSs. Once the filenames are converted into URL format on the server, the
// client must know how to take browser-generated URLs and normalize them across browsers and OSs
// into this same format for matching when consulting the cache.
//
// This method takes an input filename and converts it to a canonical format.
// - lowercase
// - full pathnames
// - forward-slashes
// - directories always terminated by slash
// - no backup indicators (e.g. "foo/../../../../bar")
// - no reset indicators (e.g. "../foo")
//
/***** NORMALIZE NAME *****/
//
public String normalizeName(String f)
{
    String full = f.replace('\\', '/').toLowerCase();

    int cut = full.indexOf("//");
    while ( cut != -1)
    {
        String part1 = full.substring(0,cut+1);
        String part2 = full.substring(cut+2);
        full = part1 + part2;
        cut = full.indexOf("//");
    }

    cut = full.indexOf("/./");
    while ( cut != -1)
    {
        String part1 = full.substring(0,cut+1);
        String part2 = full.substring(cut+3);
        full = part1 + part2;
        cut = full.indexOf("/./");
    }

    cut = full.indexOf("../");
    while ( cut != -1)
    {
        String part1 = full.substring(0,cut);

        int lsl = part1.lastIndexOf("/");

```

```

    if (lsl != -1)
        part1 = part1.substring(0,lsl);

    String part2 = full.substring(cut+3);
    full = part1 + part2;
    cut = full.indexOf("/../");
}

return full;
}
/***** MAKE FULL URL *****/
//
// This matches the current root directory with the full path spec of a file, replacing the
// root directory with the domain name that will eventually be part of the URL as seen by the
// browser. The root directory is that place in the file system where the web server will
// look for files at the domain level, i.e. the directory where www.foo.com/index.html
// will be found.
//
/***** utility method to create full URL from File at local root *****/
public String makeFullURL(File nf)
{
    String fullURL="";
    String fullfile = normalizeName(nf.getAbsolutePath());
    String rootdir = Config.getRootDir();
    if (rootdir == null)
    {
        OutMgr.print("*** ERROR - No root directory specified for configuration file *** ");
        return null;
    }
    rootdir = normalizeName(rootdir);

    if (fullfile.startsWith(rootdir))
    {
        fullURL = Config.getDomain().toLowerCase();
        fullfile = fullfile.substring( rootdir.length() );

        if ((!fullURL.endsWith("/")) && (!fullfile.startsWith("/")))
        {
            fullURL += "/";
            fullURL += fullfile;
        }
    }
    else
        OutMgr.print("*** ERROR - no match for root dir : " + fullfile );

    if (fullURL == null)
    {
        OutMgr.print("*** ERROR - root " + rootdir + " not matched with " + fullfile );
        return "ERROR-" + fullfile ;
    }

    return fullURL;
}

/***** PROCESS *****/
//
// This method navigates the file/directory structure and calls generateCoherency and generateOCD
// It is responsible for taking a file/directory spec and expanding the directories depth-first,
// recursively. The order is determined by the way the file system call returns a list of files
// in a directory.
//
/***** NAVIGATE DIRECTORY TREE TO GENERATE INFO *****/
void process(String fd, String fn)
{
    if (_server.halted())
    {
        if (!haltmsg )
        {
            OutMgr.print("-----> " +
                "Generation halted at " + new Date().toLocaleString()
                + " <-----" );
        }
    }
}

```

```

        haltmsg=true;
    }
    return;
}

//recursively traverse tree
//must have file directory fd argument

if (fd == null)
    return;

File procroot=null;

//adjust for NT vs W98
if (!fd.endsWith("/") )
    fd += "/";

if (fn != null)
    procroot = new File(fd + fn); //look at file arg to see if directory
else
    procroot = new File(fd); //if fname is null, we consider fd a directory spec.

if (procroot.isDirectory() )
{
    //process files
    String curdir = procroot.toString();
    OutMgr.print(" ");
    OutMgr.print("Processing directory: " + OutMgr.chopPath(curdir) );
    String[] flist = procroot.list();

    for (int i = 0; i<flist.length; i++)
    {
        // skip existing oc files
        String tmpf = flist[i];
        if (tmpf.length() > 3)
        {
            if ( !isOCFile(tmpf) )
                process(curdir, tmpf);
        }
    }
}
else //not a directory, process file
{
    if (fn.length() > 3)
        if (isOCFile(fn) )
            return;

    int dot = fn.lastIndexOf(".");
    if (dot > 0)
    {
        String ext = fn.substring(dot+1, fn.length()).trim() ;

        if (Config.inExtensionList(ext))
        {
            String rootdefault = Config.getDefaultFile();
            String tmpfname = normalizeName(procroot.getAbsolutePath());

            //for default files, allow all unique subobjects
            //reset global dup checker tables
            if (tmpfname.indexOf(rootdefault) != -1)
            {
                OCdupcheck = new Hashtable();
                _OCDDupcheck = new Hashtable();
            }

            //Generate the coherency information for all but the OCD file
            String newOCString = generateCoherency(fd, fn);
            String oldOCString = "";

            //We now have a new coherency string generated and we want to compare it

```

```

//to the coherency data last generated.
//If none are different, we skip producing both the OC and the OCD.

String OCfile = "oc_" + fn;
Config.setCurOutFile(OCfile);
String ocdfile = "ocd_" + fn;

FileFormatMgr old_OC = new FileFormatMgr();
File oldOCIn = new File(fd, OCfile);

try{
FileInputStream ocin = new FileInputStream(oldOCIn);
BufferedInputStream bis = new BufferedInputStream(ocin);
oldOCString = old_OC.readCoherency(bis);
}
catch (IOException ioe)
{
// OutMgr.print("      **** WARNING: old OC file not read *** " +
oldOCIn.toString() );
}

if (newOCString.compareTo(oldOCString) == 0)
{
OutMgr.print("      **** Skipping unchanged page *** " +
OutMgr.chopPath(fn) );
return;
}

//collect the index and blob in two parallel streams
ByteArrayOutputStream idxbytes = new ByteArrayOutputStream();
ByteArrayOutputStream blobbytes = new ByteArrayOutputStream();
DataOutputStream idxout = new DataOutputStream( idxbytes );
DataOutputStream blobout = new DataOutputStream( blobbytes );
//initialize the index and payload streams by inserting
//name and contents of base HTML file being processed.

try
{
int fl = (int)procroot.length();
OutMgr.print( OutMgr.FILENAMES, "processing " +
OutMgr.chopPath(tmpfname) );
String basefileURL = makeFullURL(procroot);

//then write file contents to stream
byte[] barr = new byte[fl];

FileInputStream srcin = new FileInputStream(procroot);
ByteArrayOutputStream baos = new ByteArrayOutputStream();

//write header file into start of OCD stream
//If compression is on, write header using GZIP
if (Config.getGZip())
{
GZIPOutputStream gzos = new GZIPOutputStream(baos);
while (srcin.read(barr) != -1) //compress header file into stream
gzos.write(barr);
gzos.close();
}
else
{
while (srcin.read(barr) != -1) //
baos.write(barr);
baos.close();
}

barr = baos.toByteArray();

OutMgr.print( OutMgr.FILENAMES,
"\n" + "compress base file into OCD: " +

```

```

        OutMgr.chopPath(basefileURL) +
        " size: " + fl + " compressed: " + barr.length );
    idxout.writeUTF(basefileURL); //write filename in UTF format to idx
    idxout.writeLong(barr.length); //write file size
    blobout.writeUTF(basefileURL); //write filename in UTF format to blob
    blobout.writeLong(barr.length); //write file size
    blobout.write(barr);

    OCDDupcheck.put( basefileURL, "OCD");
    srcIn.close();
}
catch (FileNotFoundException fnf)
    { OutMgr.print(" **** ERROR adding base HTML to OCD *** " + fnf.toString())
}; }

catch ( IOException ioe )
    {return;}

//now generate the OCD compilation if appropriate
generateOCD(fd, fn, idxout, blobout);

//the index and blob streams now contain what goes in the
//header and payload section of the OCD file respectively
//but have not been written to any file.
File ocdf = new File(fd, ocdfilename);

//Write the OCD file, then the OC file
//TODO: move more of the formatting knowledge to separate class FileFormatMgr
try
{
    FileOutputStream ocdfstream = new FileOutputStream(ocdf);
    FileFormatMgr ocdfformatter = new FileFormatMgr();
    ocdfformatter.writeHeader(ocdfstream);
    idxout.writeUTF(","); //delimit end of directory entries
    idxbytes.writeTo(ocdfstream); //write directory header
    blobbytes.writeTo(ocdfstream); //followed by blobs
    blobout.close();
    idxout.close();
    ocdfstream.close();
}
catch (IOException e)
    {OutMgr.print(" **** ERROR opening output file **** " + OCfile ); }

Stats._bytesWritten+=ocdf.length();
Stats._numOutFiles++;

FileFormatMgr formatter = new FileFormatMgr(fd, OCfile);
formatter.writeCoherencyFile(newOCString);

_server.updateStatLabels();
OutMgr.print(OutMgr.INCSTATUS, " Created: " + OutMgr.chopPath(ocdfilename)
    + " size: " + ocdf.length() + " bytes ");
}
}
}

/***** GENERATE OC *****/
//
/***** GENERATE OC INFO GIVEN FILE/DIR *****/
String generateCoherency(String dir, String infile)
//@input dir - directory part of filename to process
//@input infile - filename part - if null, process entire directory
//
{
    if ( (infile == null) || (infile.equals("")) )
        return null;

    if (isOCFile(infile))
        return null;
}

```



```

Config.setCurInFile(infile);

File f=new File(dir , infile);
if(!f.exists())
{
    //file not there, abort
    if (notfounds.put(infile,"NotFOUND") == null)
    {
        OutMgr.print( "      **** WARNING: Input file not found ****      ");
        OutMgr.print("      " + OutMgr.chopPath( f.getAbsolutePath() ));

        Stats._notFound++;
    }
    return null;
}

String OCREsult = "";

BufferedReader fin=null;
String stampString=null;

OutMgr.print(" ");
OutMgr.print("Input: " + OutMgr.chopPath(dir+infile) );
OutMgr.print(OutMgr.INCSTATUS, "      Size: " + f.length() );

try
{
    fin = new BufferedReader(new FileReader(f));
    //write timestamp for this file itself at top
    long msStamp=f.lastModified();
    stampString = new Long(msStamp).toString();
    String sd = makeFullURL(f) + " & %s" + stampString + " ,";
    OCREsult += sd;
}
catch (FileNotFoundException fnf)
{
    OutMgr.print(fnf + " ");
}
catch ( IOException ioe)
{
    OutMgr.print(ioe + "\n");
    return null;
}

//If OCD generation is on, assume that we have generated an OCD
//file for this HTML file
if ( Config.getOCD1() )
{
    File ocdfile = new File(dir, "ocd_" + infile);

    //disable check for OCD existence for now
    //if (ocdfile.exists())
    OCREsult += makeFullURL(ocdfile) + " & %s" + stampString + " ,";
}

dir = normalizeName(dir);
String sbit =null;
int numSRCfiles=0;
int imgsize=0;
int nestedfiles=0;
long maximgsize=0;

boolean SRCFlag=false;
sbit= readLegalString(fin);

while (sbit != null )
{
    SRCFlag=false;
    if (sbit.equalsIgnoreCase("src" ) )
    {
        SRCFlag=true;
        sbit= readLegalString(fin);
    }
}

```

```

        while ((sbit == "") && (sbit != null))
            sbit = readLegalString(fin);
        if (sbit == null)
            sbit = "EOF";
    }

    String fullpath = normalizeName(dir + sbit);
    File nf = new File(fullpath);

    int dot = sbit.lastIndexOf(".");
    String ext = sbit.substring(dot+1, sbit.length());
    if (SRCFlag && Config.inExtensionList(ext)) //Reference to HTML file - recurse
    {
        Object found = _OCdupcheck.put(fullpath, "OC"); // eliminate dups -returns null
if not there
        if (found == null)
        {
            //make sure directory path is all in left arg
            int lastslash = sbit.lastIndexOf("/");
            String fname = sbit;
            String fdir = dir;
            if (lastslash > -1)
            {
                fname = sbit.substring(lastslash+1);
                fdir = sbit.substring(0, lastslash);
                fdir = dir + fdir;
            }
            if (!fdir.endsWith("/"))
                fdir += "/";
            OutMgr.print( "OC: expanding " + fdir + " -- " + fname );

            Stats._embeddedHTML++;
            nestedfiles++;
            String tmpResult = generateCoherency(fdir, fname);
            if (tmpResult != null)
                OCResult += tmpResult;
        }
    }

    //first time we've hit this SRC= reference in this file
    //add the filename/timestamp pair to the OC string
    if ( nf.exists() && Config.isSubobject(sbit) )
    {
        Object found = _OCdupcheck.put(fullpath, "OC"); // eliminate dups -returns
null
if not there
        if (found == null)
        {
            numSRCfiles++;
            //if it's a non-html file, tally up its size

            imgsize+=nf.length();
            if (nf.length() > maximgsize)
                maximgsize = nf.length();

            sbit = makeFullURL(nf);

            long msStamp=nf.lastModified();
            stampString = new Long(msStamp).toString();

            if (sbit != null)
            {
                OutMgr.print(OutMgr.FILENAMES, "timestamp: " + OutMgr.chopPath(sbit)

                sbit += " & #" + stampString + " , " ;

                OCResult += sbit;
                Stats._numChecks++;
            }
        }
    }
};

```

```

    }
    }
    else //only report missing SRC errors once per HTML file
        if (!nf.exists() && (!sbit.startsWith("http") ))//ignore URL refs for
now
        {
            if ((notfound.put(fullpath, "NOTFOUND") == null) &&
Config.isSubobject(sbit) )
            {
                Stats._notFound++;
                OutMgr.print("      **** WARNING: subobject file not found ****
" +
                            OutMgr.chopPath(nf.getAbsolutePath() ));
            }
        }

    sbit = readLegalString(fin);
    }

    //update statistics for GUI
    Stats._bytesRead+=f.length();
    Stats._numInFiles++;
    Stats._numOutFiles++;
    _server.updateStatLabels();

    //summarize results for each file processed
    OutMgr.print(OutMgr.INCSTATUS, "      Image files: " + numSRCfiles );
    OutMgr.print(OutMgr.INCSTATUS, "      Total Image Size: " + imgsize );
    OutMgr.print(OutMgr.INCSTATUS, "      Nested HTML Files: " + nestedfiles );
    OutMgr.print(OutMgr.INCSTATUS, "      Max Image Size: " + maximgsize );

    return OCREsult;
}

/***** GENERATE OCD INFO GIVEN FILE/DIR *****/
//TODO:
// this method started as a copy of generateCoherency to preserve the common logic
// but isolate from changes temporarily. Result accumulation is structured differently from gen
OC.
//
// generateCoherency and generateOCD should be
// merged more so they share common logic. In particular the traversal results (subobject table)
// should be saved and passed from the first phase (OCD) to the second (OC)
//
/***** GENERATE OCD INFO GIVEN FILE/DIR *****/

void generateOCD(    String dir,
                    String infile,
                    DataOutputStream ios,
                    DataOutputStream dos )
//
//@input dir - input file directory
//@input infile - filename
//@input ios - output stream for accumulating index/directory results
//@input dos - output stream for accumulating data payload
//
{
    if (Config.getOCD1() == false)
        return;

    if ( (infile == null) ||
        (infile.equals("")) )
        return ;
}

```

```

if (isOCFile(infile))//TODO: - move test for OC file to a common class
    return;

File f=new File(dir, infile);
if(!f.exists())
{
    //file not there, abort
    if (notfounds.put(infile,"NotFOUND") == null)
    {
        OutMgr.print("      **** WARNING: Input file not found ****      " +
            OutMgr.chopPath(f.getAbsolutePath() ) );
        Stats._notFound++;
    }
    return;
}

BufferedReader fin=null;

try
{ fin = new BufferedReader(new FileReader(f),2048); }
catch (FileNotFoundException fnf)
{ OutMgr.print(fnf.toString() );}
catch ( IOException ioe)
{ return;}

String sbit =null;

boolean SRCFlag=false;
sbit= readLegalString(fin);

while (sbit != null )
{
    SRCFlag=false;
    if (sbit.equalsIgnoreCase("src" ) )
    {
        SRCFlag=true;
        sbit= readLegalString(fin);
        while ((sbit == "") && (sbit != null))
            sbit= readLegalString(fin);
        if (sbit == null)
            sbit = "EOF";
    }

    String fullpath = normalizeName(dir + sbit);
    File nf = new File(fullpath);

    int dot = sbit.lastIndexOf(".");
    String ext = sbit.substring(dot+1, sbit.length()) ;
    if (SRCFlag && Config.inExtensionList(ext) ) //Reference to HTML file - recurse
    {
        //make sure directory path is all in left arg
        int lastslash = sbit.lastIndexOf("/");
        String fname = sbit;
        String fdir = dir;
        if (lastslash > -1)
        {
            fname = sbit.substring(lastslash+1);
            fdir = sbit.substring(0,lastslash);
            fdir = dir + fdir; }

        if (!fdir.endsWith("/"))
            fdir += "/";

        Object found = _OCDdupcheck.put(fullpath, "OCD");

        if (found == null)
        {
            OutMgr.print("OCD: expanding  " + fdir + " -- " + fname );
        }
    }
}

```

```

        generateOCD(fdir, fname, ios, dos);
    }

    //first time we've hit this SRC= reference in this file
    //append records and binary information to OCD_ file for this HTML file
    //We also need to ensure that an OCD_ entry appears in the OC_ string
    if ( nf.exists() && Config.inSubobjectExtensionList(ext) )
    {
        Object found = _OCDdupcheck.put(fullpath, "OCD");
        if (found == null)
        {
            try{
                int fl = (int)nf.length();
                sbit = makeFullURL(nf);
                OutMgr.print(OutMgr.FILENAMES, "add object: " + OutMgr.chopPath(sbit)
+ " size: " + fl );

                ios.writeUTF(sbit); //write filename in UTF format to idx
                ios.writeLong(fl); //write file size
                dos.writeUTF(sbit); //write filename in UTF format to blob
                dos.writeLong(fl); //write file size

                Stats._bytesWritten += sbit.length()*2 + 8;

                //then write file contents to stream
                byte[] barr = new byte[fl];

                FileInputStream srcin = new FileInputStream(nf);
                while (srcin.read(barr) != -1)
                    dos.write(barr);

                Stats._bytesRead += fl;
                server.updateStatLabels();
            }//end try
            catch (IOException ex)
            {
                OutMgr.print("      **** WARNING: exception reading file " +
                OutMgr.chopPath(sbit) + ex + " **** ");
            }
        }
    } //end creating ocd leaf entry

    sbit = readLegalString(fin);
} //end while loop

} //end generateOCD

/***** DELETED OC INFO *****/
//TODO: move this method to a class that encapsulates the functionality
// and call it from process() instead of replicating the directory traversal logic
//
/***** DELETED OC INFO *****/
void deleteOCFiles(String fd, String fn)
{
    boolean ok;
    //recursively traverse tree

    if (fd == null)
        return;
    File procroot=null;
    //adjust for NT vs W98
    if (!fd.endsWith("\\"))
        fd += "\\";

    if (fn != null)
        procroot = new File(fd + fn); //look at file arg to see if directory
    else
        procroot = new File(fd);
}

```

```
//if (procroot != null)
if (procroot.isDirectory() )
    //process files
    String curdir = procroot.getAbsolutePath();
    OutMgr.print( "Cleaning directory: " + OutMgr.chopPath(curdir) );
    String[] flist = procroot.list();

    for (int i = 0; i<flist.length; i++)
        deleteOCFiles(curdir, flist[i]);
}
else
{
    if ( isOCFile(fn))
    {
        ok = procroot.exists();
        OutMgr.print("      Deleting " + OutMgr.chopPath(fn) );
        ok = procroot.delete();
        Stats._numInFiles++;
    }
}
}
//{{DECLARE_CONTROLS
//}}
}
```

```

import java.io.*;
import java.awt.TextArea;
import java.util.Vector;

public class OutMgr
{
    public static boolean    TTOUT=true;

    public static int        FILENAMES=1;
    public static boolean    PRINT_FILENAMES=false;

    public static int        DEBUGMSG=2;
    public static boolean    PRINT_DEBUGMSG= false;

    public static int        INCSTATUS=3;
    public static boolean    PRINT_INCSTATUS= false;

    private static boolean    FILEOUT=false;
    public static boolean    PRINT_FULLNAMES=false;

    public static boolean    LOG_ALWAYS=false;
    public static boolean    LOG_APPEND=false;

    public static TextArea    textout=null;
    private static PrintStream ps = null;
    private static int    outbytes=0;
    private static int    MaxLen=50;

    private static String    _logfilename;
    private static String    _warningString = " **** WARNING: Previous output truncated. Use logfile
to capture full trace. \n\n";
    private static Vector    _stringBuf=new Vector(3000);

    public static void setTextArea( TextArea ta)
    {textout = ta; }

    public static void on()
    {TTOUT=true;}

    public static void off()
    {TTOUT=false;}

    public static String chopPath(String inpath)
    {if (PRINT_FULLNAMES)
        return chopPath(inpath,inpath.length());
        else
        return chopPath(inpath,MaxLen); }

    public static String chopPath(String inpath,int maxlen)
    {
        int l = inpath.length();
        int diff = 0;
        if (l > maxlen)
            diff = l-maxlen;
        else
            return inpath;

        //we need to nip out diff characters from the middle
        int start = maxlen/3;
        int end = maxlen - start;
        String firstpart = inpath.substring(0,start);
        String lastpart = inpath.substring(l-end);
        String newpath = firstpart + "..." + lastpart;
        return newpath;
    }

    public static void forceprint(String msg)
    {
        if (FILEOUT == true)

```

```

        ps.print(msg );
    }

    public static void setLogfile(String lfname)
    { _logfile = lfname; }

    public static void setLogAppend(boolean tf)
    { LOG_APPEND = tf; }

    public static void closeLog()
    { if (ps != null)
      ps.close();
    }

    public static void setLogOn()
    {
        try
        {
            ps = new PrintStream(new FileOutputStream(_logfile, LOG_APPEND ));
            FILEOUT=true;
        }
        catch(IOException e)
        { System.out.println("error opening file: " + e);}
    }

    public static void setLogOff()
    {
        if (ps != null)
            ps.close();
        FILEOUT=false;
    }

    public static boolean fileOut()
    { return FILEOUT; }

    public static void print(int type, String msg)
    {
        if ((type == FILENAMES) && (PRINT_FILENAMES))
            print(msg);

        if ((type == DEBUGMSG) && (PRINT_DEBUGMSG))
            print(msg);

        if ((type == INCSTATUS) && (PRINT_INCSTATUS))
            print(msg);
    }

    //unconditional print
    public static void print( String msg)
    {
        textout.append(msg + "\n");
        outbytes += msg.length();

        if (outbytes > 26000)
        {
            textout.replaceRange(_warningString, 0 , 1000);
            outbytes= outbytes-(1000 + _warningString.length ());
        }

        if (FILEOUT == true)
            ps.print( msg + "\r\n" );
    }
}

```



```

import java.awt.*;
import symantec.itools.awt.util.Calendar;
import symantec.itools.awt.BorderPanel;

public class ProcessingDialog extends Dialog {
    Frame _parent;
    public ProcessingDialog(Frame parent, boolean modal)
    {
        super(parent, modal);
        _parent=parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        {{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(253,320);
        setVisible(false);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setBounds(46,294,64,20);
        cancelButton.setLabel("Cancel");
        add(cancelButton);
        cancelButton.setBounds(110,294,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setLabel(" Object Packing ");
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingRight(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingBottom(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingTop(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setPaddingLeft(0);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(14,12,230,163);
        checkbox7.setLabel("OCDPPD");
        borderPanell.add(checkbox7);
        checkbox7.setFont(new Font("Dialog", Font.PLAIN, 12));
        checkbox7.setBounds(17,75,164,24);
        checkbox5.setLabel("OCD144");
        borderPanell.add(checkbox5);
        checkbox5.setFont(new Font("Dialog", Font.PLAIN, 12));
        checkbox5.setBounds(17,59,180,24);
        }}}
    }
}

```

```

checkbox4.setLabel("OCD33");
borderPanel1.add(checkbox4);
checkbox4.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox4.setBounds(17, 39, 180, 24);
checkbox3.setLabel("OCD56");
borderPanel1.add(checkbox3);
checkbox3.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox3.setBounds(17, 23, 180, 24);
OCD1Checkbox.setLabel("OCD ");
borderPanel1.add(OCD1Checkbox);
OCD1Checkbox.setFont(new Font("Dialog", Font.PLAIN, 12));
OCD1Checkbox.setBounds(17, 3, 140, 24);
ocdtvlabel.setLabel("OCDTV");
borderPanel1.add(ocdtvlabel);
ocdtvlabel.setBounds(17, 95, 100, 24);
compressCheckbox.setLabel("Compression ");
borderPanel1.add(compressCheckbox);
compressCheckbox.setBounds(17, 111, 132, 30);
try {
    borderPanel2.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setLabel(" Special Formats ");
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setPaddingRight(0);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setPaddingBottom(0);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setPaddingTop(0);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setPaddingLeft(0);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel2.setLayout(null);
add(borderPanel2);
borderPanel2.setBounds(12, 188, 230, 100);
checkbox6.setLabel(".GIF Graphics");
borderPanel2.add(checkbox6);
checkbox6.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox6.setBounds(19, 1, 100, 24);
checkbox8.setLabel(".JPG Graphics");
borderPanel2.add(checkbox8);
checkbox8.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox8.setBounds(19, 25, 100, 24);
checkbox9.setLabel(".ASP Scripts");
borderPanel2.add(checkbox9);
checkbox9.setFont(new Font("Dialog", Font.PLAIN, 12));
checkbox9.setBounds(19, 49, 100, 24);
button2.setLabel("Apply");
add(button2);
button2.setBounds(175, 294, 64, 20);
setTitle("WWWhoosh Processing");
//}}

OCD1Checkbox.setState(Config.getOCD1());
compressCheckbox.setState(Config.getGZip());

```

```

    //{{{REGISTER_LISTENERS
    SymWindow aSymWindow = new SymWindow();
    this.addWindowListener(aSymWindow);
    SymAction lSymAction = new SymAction();
    okButton.addActionListener(lSymAction);
    button2.addActionListener(lSymAction);
    cancelButt.addActionListener(lSymAction);
    //}}}

}

public ProcessingDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }

    super.setVisible(b);
}

//{{{DECLARE_CONTROLS
java.awt.Button okButton = new java.awt.Button();
java.awt.Button cancelButt = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
java.awt.Checkbox checkbox7 = new java.awt.Checkbox();
java.awt.Checkbox checkbox5 = new java.awt.Checkbox();
java.awt.Checkbox checkbox4 = new java.awt.Checkbox();
java.awt.Checkbox checkbox3 = new java.awt.Checkbox();
java.awt.Checkbox OCD1Checkbox = new java.awt.Checkbox();
java.awt.Checkbox ocdivlabel = new java.awt.Checkbox();
java.awt.Checkbox compressCheckbox = new java.awt.Checkbox();

```

```

symantec.itools.awt.BorderPanel borderPanel2 = new symantec.itools.awt.BorderPanel();
java.awt.Checkbox checkbox6 = new java.awt.Checkbox();
java.awt.Checkbox checkbox8 = new java.awt.Checkbox();
java.awt.Checkbox checkbox9 = new java.awt.Checkbox();
java.awt.Button button2 = new java.awt.Button();
//))

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
        else if (object == button2)
            button2_ActionPerformed(event);
        else if (object == cancelButt)
            cancelButt_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    if (compressCheckbox.getState())
        Config.setGZip(true);
    else
        Config.setGZip(false);

    if (OCD1Checkbox.getState())
        Config.setOCD1(true);
    else
        Config.setOCD1(false);

    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowOpened(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == ProcessingDialog.this)
            ProcessingDialog_WindowOpened(event);
    }

    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == ProcessingDialog.this)
            ProcessingDialog_WindowClosing(event);
    }
}

```

```
void ProcessingDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.
    ConfirmSaveDialog csd = null;

    try {
        // ConfirmSaveDialog Create and show as modal
        csd = new ConfirmSaveDialog( _parent, true);
        csd.setVisible(true);
    }
    catch (Exception e) { }

    if (csd.SAVEIT)
        Config.setOCD1(OCD1Checkbox.getState());

    setVisible(false);
}

void button2_ActionPerformed(java.awt.event.ActionEvent event)
{
    if (compressCheckbox.getState())
        Config.setGZip(true);
    else
        Config.setGZip(false);

    if (OCD1Checkbox.getState())
        Config.setOCD1(true);
    else
        Config.setOCD1(false);
}

void ProcessingDialog_WindowOpened(java.awt.event.WindowEvent event)
{
}

void cancelButt_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    setVisible(false);
}
}
```

```

/*
 * A basic extension of the java.awt.Dialog class
 */

import java.awt.*;
import java.awt.event.*;
import symantec.itools.awt.BorderPanel;

public class QuitDialog extends Dialog
{
    public QuitDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        //Keep a local reference to the invoking frame
        frame = parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.
        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(249,117);
        setVisible(false);
        yesButton.setLabel(" Yes ");
        add(yesButton);
        yesButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        yesButton.setBounds(107,87,64,20);
        noButton.setLabel(" No ");
        add(noButton);
        noButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        noButton.setBounds(171,87,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(8,7,237,80);
        label1.setText("Save Config changes before exit?");
        label1.setAlignment(java.awt.Label.CENTER);
        borderPanell.add(label1);
        label1.setBounds(5,23,202,23);
        setTitle("Exit WWWhoosh");
        //}}}

        //{{{REGISTER_LISTENERS
        SymWindow aSymWindow = new SymWindow();
        this.addWindowListener(aSymWindow);
        SymAction lSymAction = new SymAction();
        noButton.addActionListener(lSymAction);
        yesButton.addActionListener(lSymAction);
        //}}}
    }

    public void addNotify()
    {
        // Record the size of the window prior to calling parents addNotify.
        Dimension d = getSize();

        super.addNotify();

        if (fComponentsAdjusted)
            return;

        // Adjust components according to the insets

```

```

        setSize(getInsets().left + getInsets().right + d.width, getInsets().top +
getInsets().bottom + d.height);
        Component components[] = getComponents();
        for (int i = 0; i < components.length; i++)
        {
            Point p = components[i].getLocation();
            p.translate(getInsets().left, getInsets().top);
            components[i].setLocation(p);
        }
        fComponentsAdjusted = true;
    }

    public QuitDialog(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    /**
     * Shows or hides the component depending on the boolean flag b.
     * @param b if true, show the component; otherwise, hide the component.
     * @see java.awt.Component#isVisible
     */
    public void setVisible(boolean b)
    {
        if(b)
        {
            Rectangle bounds = getParent().getBounds();
            Rectangle abounds = getBounds();

            setLocation(bounds.x + (bounds.width - abounds.width)/ 2,
                bounds.y + (bounds.height - abounds.height)/2);
            Toolkit.getDefaultToolkit().beep();
        }
        super.setVisible(b);
    }

    // Used for addNotify check.
    boolean fComponentsAdjusted = false;
    // Invoking frame
    Frame frame = null;

    /**{DECLARE_CONTROLS
    java.awt.Button yesButton = new java.awt.Button();
    java.awt.Button noButton = new java.awt.Button();
    symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
    java.awt.Label labell = new java.awt.Label();
    /**}}

    class SymAction implements java.awt.event.ActionListener
    {
        public void actionPerformed(java.awt.event.ActionEvent event)
        {
            Object object = event.getSource();
            if (object == yesButton)
                yesButton_ActionPerformed(event);
            else if (object == noButton)
                noButton_ActionPerformed(event);
        }
    }

    void yesButton_ActionPerformed(java.awt.event.ActionEvent event)
    {
        // to do: code goes here.

        yesButton_ActionPerformed_Interaction1(event);
    }

```

```
void yesButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        frame.setVisible(false);    // Hide the invoking frame
        frame.dispose();            // Free system resources
        this.dispose();             // Free system resources
        System.exit(0);             // close the application
    } catch (Exception e) {
    }
}
```

```
void noButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    noButton_ActionPerformed_Interaction1(event);
}
```

```
void noButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}
```

```
class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == QuitDialog.this)
            QuitDialog_WindowClosing(event);
    }
}
```

```
void QuitDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.

    QuitDialog_WindowClosing_Interaction1(event);
}
```

```
void QuitDialog_WindowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

}
```



```

import java.awt.*;
import symantec.itools.awt.util.Calendar;
import symantec.itools.awt.BorderPanel;

public class SchedulingDialog extends Dialog {
    Frame _parent=null;
    public SchedulingDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        _parent=parent;

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(264,196);
        setVisible(false);
        okButton.setLabel("OK");
        add(okButton);
        okButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        okButton.setBounds(60,168,64,20);
        button1.setLabel("Cancel");
        add(button1);
        button1.setFont(new Font("Dialog", Font.PLAIN, 12));
        button1.setBounds(122,168,64,20);
        applyButton.setLabel("Apply");
        add(applyButton);
        applyButton.setFont(new Font("Dialog", Font.PLAIN, 12));
        applyButton.setBounds(185,168,64,20);
        try {
            borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setLabel(" Scheduling ");
        }
        catch(java.beans.PropertyVetoException e) { }
        try {
            borderPanell.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
        }
        catch(java.beans.PropertyVetoException e) { }
        borderPanell.setLayout(null);
        add(borderPanell);
        borderPanell.setBounds(4,12,256,156);
        manualCheckbox.setLabel("Manually");
        borderPanell.add(manualCheckbox);
        manualCheckbox.setFont(new Font("Dialog", Font.PLAIN, 12));
        manualCheckbox.setBounds(74,10,138,21);
        borderPanell.add(textField1);
        textField1.setBounds(74,72,132,18);
        borderPanell.add(textField2);
        textField2.setBounds(74,96,132,18);
        label1.setText("Start:");
        borderPanell.add(label1);
        label1.setBounds(40,68,29,20);
        label3.setText("Stop:");
        borderPanell.add(label3);
        label3.setBounds(41,92,28,19);
        label4.setText("Interval (min):");
        borderPanell.add(label4);
        label4.setBounds(0,44,71,23);
        minutesText.setText("1");
        minutesText.setEnabled(false);
    }
}

```

```

borderPanel1.add(minutesText);
minutesText.setBounds(75,49,132,18);
setTitle("WWWhoosh Scheduling");
//}}

//{{REGISTER_LISTENERS
SymWindow aSymWindow = new SymWindow();
this.addWindowListener(aSymWindow);
SymAction lSymAction = new SymAction();
okButton.addActionListener(lSymAction);

button1.addActionListener(lSymAction);
applyButton.addActionListener(lSymAction);
SymItem lSymItem = new SymItem();
manualCheckbox.addItemListener(lSymItem);
//}}

}

public SchedulingDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    // Only do this once.
    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    Insets insets = getInsets();
    setSize(insets.left + insets.right + d.width, insets.top + insets.bottom + d.height);
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {
        Point p = components[i].getLocation();
        p.translate(insets.left, insets.top);
        components[i].setLocation(p);
    }

    // Used for addNotify check.
    fComponentsAdjusted = true;
}

public void setVisible(boolean b)
{
    if (b)
    {
        Rectangle bounds = getParent().getBounds();
        Rectangle abounds = getBounds();

        setLocation(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);
    }

    super.setVisible(b);
}

//{{DECLARE_CONTROLS
java.awt.Button okButton = new java.awt.Button();
java.awt.Button button1 = new java.awt.Button();
java.awt.Button applyButton = new java.awt.Button();

```

```

symantec.itools.awt.BorderPanel borderPanell = new symantec.itools.awt.BorderPanel();
java.awt.Checkbox manualCheckbox = new java.awt.Checkbox();
java.awt.TextField textField1 = new java.awt.TextField();
java.awt.TextField textField2 = new java.awt.TextField();
java.awt.Label label1 = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
java.awt.Label label4 = new java.awt.Label();
java.awt.TextField minutesText = new java.awt.TextField();
//}}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == okButton)
            okButton_ActionPerformed(event);
        if (object == button1)
            button1_ActionPerformed(event);
        else if (object == applyButton)
            applyButton_ActionPerformed(event);
    }
}

void okButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    applyButton_ActionPerformed(event);
    okButton_ActionPerformed_Interaction1(event);
}

void okButton_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
    } catch (Exception e) {
    }
}

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowOpened(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == SchedulingDialog.this)
            SchedulingDialog_WindowOpened(event);
    }

    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == SchedulingDialog.this)
            SchedulingDialog_WindowClosing(event);
    }
}

void SchedulingDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    ConfirmSaveDialog csd = null;

    try {
        // ConfirmSaveDialog Create and show as modal
        csd = new ConfirmSaveDialog(_parent, true);
        csd.setVisible(true);
    }
}

```

```

    }
    catch (Exception e) { }

    if (csd.SAVEIT)
    {
        Config.setContinuous( !manualCheckbox.getState() );
        Integer geninterval = new Integer(minutesText.getText() );
        Config.setGenInterval(geninterval.intValue() );
    }
    setVisible(false);
}

void button1_ActionPerformed(java.awt.event.ActionEvent event)
{
    try {
        // ProcessingDialog Hide the ProcessingDialog
        this.setVisible(false);
        this.dispose();
    } catch (Exception e) {
    }
}

void SchedulingDialog_WindowOpened(java.awt.event.WindowEvent event)
{
    // to do: code goes here.
    manualCheckbox.setState(!Config.getContinuous());
    if (Config.getContinuous())
        minutesText.enable();
}

void applyButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    Config.setContinuous( !manualCheckbox.getState() );
    Integer geninterval = new Integer(minutesText.getText() );
    Config.setGenInterval(geninterval.intValue() );
}

class SymItem implements java.awt.event.ItemListener
{
    public void itemStateChanged(java.awt.event.ItemEvent event)
    {
        Object object = event.getSource();
        if (object == manualCheckbox)
            manualCheckbox_ItemStateChanged(event);
    }
}

void manualCheckbox_ItemStateChanged(java.awt.event.ItemEvent event)
{
    if ( manualCheckbox.getState() )
        minutesText.disable();
    else
        minutesText.enable();
}
}

```

```
import java.io.*;
import java.util.Date;

public class Stats
{
    // The following function is a placeholder for control initialization.
    // You should call this function from a constructor or initialization function.
    public void vcInit() {
        //{{{INIT_CONTROLS
        //}}}
    }

    private static PrintStream outStream = System.out;

    public static int _maxDepth=10;
    public static int _curDepth=0;
    public static int _numInFiles=0;
    public static int _numOutFiles=0;
    public static int _numChecks=0;
    public static int _notFound=0;
    public static int _bytesRead=0;
    public static int _bytesWritten=0;
    public static int _embeddedHTML=0;

    public static void clear()
    {
        _embeddedHTML=0;
        _maxDepth=10;
        _curDepth=0;
        _numInFiles=0;
        _numOutFiles=0;
        _numChecks=0;
        _notFound=0;
        _bytesRead=0;
        _bytesWritten=0;
    }

    //{{{DECLARE_CONTROLS
    //}}}
}
```

```

/*
    This simple extension of the java.awt.Frame class
    contains all the elements necessary to act as the
    main window of an application.
*/

import java.awt.*;
import java.util.*;
import java.util.zip.*;
import java.io.*;
import java.awt.image.ImageObserver;
import symantec.itools.awt.StatusBar;
import symantec.itools.awt.BorderPanel;
import symantec.itools.awt.Label3D;
import com.sun.java.swing.JTabbedPane;
import com.sun.java.swing.JTextArea;
import com.symantec.itools.swing.models.UnsignedNumberDocument;
import com.sun.java.swing.JList;
import com.sun.java.swing.JScrollPane;
import com.symantec.itools.swing.borders.BevelBorder;
import com.symantec.itools.swing.borders.EtchedBorder;
import com.symantec.itools.swing.borders.SoftBevelBorder;
import com.symantec.itools.swing.models.StringListModel;

/***** WWWhoosh Server *****/
//
// @author Mark Palmer
// @version 1.9
//
// Modifications:
//
// Build      Date      Description
//
// 1.9        18-Aug-1999    - Hack Text Output area to recycle its memory
//
//                                - add append-log checkbox and capability, modify INI, buttons
//
//
/***** WWWhoosh Server GUI *****/

public class WWWhooshServer extends Frame
{
    public WWWhooshServer()
    {
        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches
        // what Visual Cafe can generate, or Visual Cafe may be unable to back
        // parse your Java file into its visual environment.

        Config.loadLast();
        dateLabel.setText(Config.getDate());
        domainLabel.setText(Config.getDomain());
        configLabel.setText(Config.getFile());

        //{{{INIT_CONTROLS
        setLayout(null);
        setBackground(java.awt.Color.lightGray);
        setSize(496,483);
        setVisible(false);
        button4.setLabel("Clear");
        add(button4);
        button4.setBounds(345,449,64,20);
        logbutt.setLabel("Log File");
        add(logbutt);
        logbutt.setBounds(409,449,64,20);
        genButton.setLabel("Generate");
    }
}

```

```

add(genButton);
genButton.setBounds(406,115,64,20);
stopButt.setLabel("Stop");
add(stopButt);
stopButt.setBounds(341,115,64,20);
RmButton.setLabel("Remove");
add(RmButton);
RmButton.setBounds(276,115,64,20);
try {
    borderPanell.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanell.setLabel(" Run Statistics ");
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanell.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
}
catch(java.beans.PropertyVetoException e) { }
borderPanell.setLayout(null);
add(borderPanell);
borderPanell.setBounds(17,131,470,108);
label13.setText(" Created:");
borderPanell.add(label13);
label13.setFont(new Font("Dialog", Font.PLAIN, 12));
label13.setBounds(15,23,60,15);
label11.setText(" Checks:");
borderPanell.add(label11);
label11.setBounds(14,102,60,24);
label9.setText(" Files: ");
borderPanell.add(label9);
label9.setBounds(14,78,60,24);
label10.setText(" Errors:");
borderPanell.add(label10);
label10.setBounds(14,126,60,24);
label4.setText(" Read:");
borderPanell.add(label4);
label4.setBounds(15,3,58,19);
label16.setText(" Bytes Written:");
borderPanell.add(label16);
label16.setBounds(242,21,78,19);
label15.setText(" Bytes Read:");
borderPanell.add(label15);
label15.setBounds(242,2,73,21);
bytesOutputLabel.setText("size of outputs");
borderPanell.add(bytesOutputLabel);
bytesOutputLabel.setBounds(329,24,99,16);
errorLabel.setText("files not found");
borderPanell.add(errorLabel);
errorLabel.setBounds(76,40,95,19);
bytesReadLabel.setText("total of files input");
borderPanell.add(bytesReadLabel);
bytesReadLabel.setBounds(329,7,99,16);
label5.setText("Errors:");
borderPanell.add(label5);
label5.setBounds(19,39,53,20);
fileOpenedLabel.setText("files opened");
borderPanell.add(fileOpenedLabel);
fileOpenedLabel.setBounds(76,3,95,19);
numCheckLabel.setText("graphics files");
borderPanell.add(numCheckLabel);
numCheckLabel.setBounds(329,41,99,16);
OCCCountLabel.setText("files created");
borderPanell.add(OCCCountLabel);
OCCCountLabel.setFont(new Font("Dialog", Font.PLAIN, 12));
OCCCountLabel.setBounds(76,22,95,19);
label12.setText("text");
borderPanell.add(label12);

```

```

label12.setBounds(254,30,8,8);
label14.setText("Subobjects:");
borderPanel1.add(label14);
label14.setBounds(245,40,73,19);
try {
    borderPanel2.setBevelStyle(symantec.itools.awt.BorderPanel.BEVEL_RAISED);
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setLabel(" Configuration ");
}
catch(java.beans.PropertyVetoException e) { }
try {
    borderPanel2.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
}
catch(java.beans.PropertyVetoException e) { }
borderPanel2.setLayout(null);
add(borderPanel2);
borderPanel2.setBounds(16,8,471,109);
borderPanel2.add(dateLabel);
dateLabel.setBounds(321,44,132,16);
label21.setText(" Checks:");
borderPanel2.add(label21);
label21.setBounds(14,102,60,24);
label22.setText(" Files: ");
borderPanel2.add(label22);
label22.setBounds(14,78,60,24);
label24.setText(" Errors:");
borderPanel2.add(label24);
label24.setBounds(14,126,60,24);
label25.setText("total unresolved SRC tags");
borderPanel2.add(label25);
label25.setBounds(74,126,180,22);
label18.setText(" Output:");
borderPanel2.add(label18);
label18.setBounds(15,23,56,18);
label11.setText(" Input:");
borderPanel2.add(label11);
label11.setBounds(15,4,56,16);
outFileLabel.setText("current output file");
borderPanel2.add(outFileLabel);
outFileLabel.setBounds(76,23,160,18);
inFileLabel.setText("current input file");
borderPanel2.add(inFileLabel);
inFileLabel.setBounds(76,4,160,16);
label3.setText("Optimization: ");
borderPanel2.add(label3);
label3.setBounds(242,23,74,18);
label6.setText("WWWwhoosh Level 1");
borderPanel2.add(label6);
label6.setBounds(321,23,124,18);
label2.setText("Domain: ");
borderPanel2.add(label2);
label2.setBounds(242,4,75,16);
borderPanel2.add(domainLabel);
domainLabel.setBounds(321,4,124,16);
label17.setText("Last Run: ");
borderPanel2.add(label17);
label17.setBounds(242,44,72,16);
label7.setText(" Config: ");
borderPanel2.add(label7);
label7.setBounds(15,43,56,16);
configLabel.setText("configuration file");
borderPanel2.add(configLabel);
configLabel.setBounds(76,43,157,16);
add(textAreal);
textAreal.setBounds(24,244,457,193);
setTitle("WWWwhoosh Site Optimizer");
setResizable(false);

```



```

    ///}

    OutMgr.setTextArea(textArea1);
    OutMgr.on();

    //{{{INIT_MENUS
    menu1.setLabel("File");
    menu1.add(newMenuItem);
    newMenuItem.setLabel("New");
    newMenuItem.setShortcut(new MenuShortcut(java.awt.event.KeyEvent.VK_N, false));
    menu1.add(openMenuItem);
    openMenuItem.setLabel("Load Config...");
    openMenuItem.setShortcut(new MenuShortcut(java.awt.event.KeyEvent.VK_O, false));
    menu1.add(saveMenuItem);
    saveMenuItem.setLabel("Save Config");
    saveMenuItem.setShortcut(new MenuShortcut(java.awt.event.KeyEvent.VK_S, false));
    menu1.add(saveAsMenuItem);
    saveAsMenuItem.setLabel("Save Config As...");
    menu1.add(separatorMenuItem);
    separatorMenuItem.setLabel("-");
    menu1.add(exitMenuItem);
    exitMenuItem.setLabel("Exit");
    mainMenuBar.add(menu1);
    menu2.setLabel("Edit");
    menu2.add(inputsmenuitem);
    inputsmenuitem.setLabel("Inputs");
    menu2.add(processingmenuitem);
    processingmenuitem.setLabel("Processing");
    menu2.add(schedmenuitem);
    schedmenuitem.setLabel("Scheduling");
    menu2.add(logLevelMenuItem);
    logLevelMenuItem.setLabel("Logging");
    mainMenuBar.add(menu2);
    menu3.setLabel("Help");
    menu3.add(aboutMenuItem);
    aboutMenuItem.setLabel("About...");
    mainMenuBar.add(menu3);
    //$$ mainMenuBar.move(30, 543);
    setMenuBar(mainMenuBar);
    ///}

    //{{{REGISTER_LISTENERS
    SymWindow aSymWindow = new SymWindow();
    this.addWindowListener(aSymWindow);

    SymAction lSymAction = new SymAction();

    openMenuItem.addActionListener(lSymAction);
    exitMenuItem.addActionListener(lSymAction);
    aboutMenuItem.addActionListener(lSymAction);
    SymComponent aSymComponent = new SymComponent();
    inputsmenuitem.addActionListener(lSymAction);
    processingmenuitem.addActionListener(lSymAction);
    schedmenuitem.addActionListener(lSymAction);
    genButton.addActionListener(lSymAction);
    button4.addActionListener(lSymAction);
    domainLabel.addComponentListener(aSymComponent);
    saveMenuItem.addActionListener(lSymAction);
    RmButton.addActionListener(lSymAction);
    stopButt.addActionListener(lSymAction);

    saveAsMenuItem.addActionListener(lSymAction);
    logbutt.addActionListener(lSymAction);
    LogLevelMenuItem.addActionListener(lSymAction);
    ///}
    try {
        FileOutputStream bld = new FileOutputStream("Build.java");
        String verstring = "public class Build {public static int VERSION=2; public static int

```

```

NUMBER=" + (Build.NUMBER+1) + " "; }";
    bld.write(verstring.getBytes());
    bld.close();
}
catch (IOException ioe)
{}
}

public WWWhooshServer(String title)
{
    this();
    setTitle(title);
}
private boolean _halt=false;
public synchronized void halt()
{ _halt=true; }

public synchronized boolean halted()
{
    return _halt;
}

/**
 * Shows or hides the component depending on the boolean flag b.
 * @param b if true, show the component; otherwise, hide the component.
 * @see java.awt.Component#isVisible
 */
public void setVisible(boolean b)
{
    if(b)
    {
        setLocation(50, 50);
    }
    super.setVisible(b);
}

static public void main(String args[])
{
    try
    {
        //Create a new instance of our application's frame, and make it visible.
        (new WWWhooshServer()).setVisible(true);
    }
    catch (Throwable t)
    {
        System.err.println(t);
        t.printStackTrace();
        //Ensure the application exits with an error condition.
        System.exit(1);
    }
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    setSize(getInsets().left + getInsets().right + d.width, getInsets().top +
getInsets().bottom + d.height);
}

```

```

Component components[] = getComponents();
for (int i = 0; i < components.length; i++)
{
    Point p = components[i].getLocation();
    p.translate(getInsets().left, getInsets().top);
    components[i].setLocation(p);
}
fComponentsAdjusted = true;
}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

int maxplen = 40;
Optimizer _opt=null;

//{{DECLARE_CONTROLS
java.awt.Button button4 = new java.awt.Button();
java.awt.Button logbutt = new java.awt.Button();
java.awt.Button genButton = new java.awt.Button();
java.awt.Button stopButt = new java.awt.Button();
java.awt.Button RmButton = new java.awt.Button();
symantec.itools.awt.BorderPanel borderPanel1 = new symantec.itools.awt.BorderPanel();
java.awt.Label label13 = new java.awt.Label();
java.awt.Label label11 = new java.awt.Label();
java.awt.Label label9 = new java.awt.Label();
java.awt.Label label10 = new java.awt.Label();
java.awt.Label label4 = new java.awt.Label();
java.awt.Label label16 = new java.awt.Label();
java.awt.Label label15 = new java.awt.Label();
java.awt.Label bytesOutputLabel = new java.awt.Label();
java.awt.Label errorLabel = new java.awt.Label();
java.awt.Label bytesReadLabel = new java.awt.Label();
java.awt.Label label5 = new java.awt.Label();
java.awt.Label fileOpenedLabel = new java.awt.Label();
java.awt.Label numCheckLabel = new java.awt.Label();
java.awt.Label OCCountLabel = new java.awt.Label();
java.awt.Label label12 = new java.awt.Label();
java.awt.Label label14 = new java.awt.Label();
symantec.itools.awt.BorderPanel borderPanel2 = new symantec.itools.awt.BorderPanel();
java.awt.Label dateLabel = new java.awt.Label();
java.awt.Label label21 = new java.awt.Label();
java.awt.Label label22 = new java.awt.Label();
java.awt.Label label24 = new java.awt.Label();
java.awt.Label label25 = new java.awt.Label();
java.awt.Label label8 = new java.awt.Label();
java.awt.Label label1 = new java.awt.Label();
java.awt.Label outFileLabel = new java.awt.Label();
java.awt.Label inFileLabel = new java.awt.Label();
java.awt.Label label3 = new java.awt.Label();
java.awt.Label label6 = new java.awt.Label();
java.awt.Label label2 = new java.awt.Label();
java.awt.Label domainLabel = new java.awt.Label();
java.awt.Label label17 = new java.awt.Label();
java.awt.Label label7 = new java.awt.Label();
java.awt.Label configLabel = new java.awt.Label();
java.awt.TextArea textArea1 = new java.awt.TextArea();
//}}

//{{DECLARE_MENUS
java.awt.MenuBar mainMenuBar = new java.awt.MenuBar();
java.awt.Menu menu1 = new java.awt.Menu();
java.awt.MenuItem newMenuItem = new java.awt.MenuItem();
java.awt.MenuItem openMenuItem = new java.awt.MenuItem();
java.awt.MenuItem saveMenuItem = new java.awt.MenuItem();
java.awt.MenuItem saveAsMenuItem = new java.awt.MenuItem();
java.awt.MenuItem separatorMenuItem = new java.awt.MenuItem();
java.awt.MenuItem exitMenuItem = new java.awt.MenuItem();
java.awt.Menu menu2 = new java.awt.Menu();

```

```

java.awt.MenuItem inputsmenuitem = new java.awt.MenuItem();
java.awt.MenuItem processingmenuitem = new java.awt.MenuItem();
java.awt.MenuItem schedmenuitem = new java.awt.MenuItem();
java.awt.MenuItem LogLevelMenuItem = new java.awt.MenuItem();
java.awt.Menu menu3 = new java.awt.Menu();
java.awt.MenuItem aboutMenuItem = new java.awt.MenuItem();
//))

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == WWWhooshServer.this)
            WWWhooshServer_windowClosing(event);
    }
}

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == openMenuItem)
            openMenuItem_ActionPerformed(event);
        else if (object == aboutMenuItem)
            aboutMenuItem_ActionPerformed(event);
        else if (object == exitMenuItem)
            exitMenuItem_ActionPerformed(event);
        else if (object == inputsmenuitem)
            inputsmenuitem_ActionPerformed(event);
        else if (object == processingmenuitem)
            processingmenuitem_ActionPerformed(event);

        else if (object == schedmenuitem)
            schedmenuitem_ActionPerformed(event);
        else if (object == genButton)
            genButton_ActionPerformed(event);
        else if (object == button4)
            button4_ActionPerformed(event);
        if (object == saveMenuItem)
            saveMenuItem_ActionPerformed(event);
        else if (object == RmButton)
            RmButton_ActionPerformed(event);
        else if (object == stopButt)
            stopButt_ActionPerformed(event);
        else if (object == saveAsMenuItem)
            saveAsMenuItem_ActionPerformed(event);
        else if (object == logbutt)
            logbutt_ActionPerformed(event);
        else if (object == LogLevelMenuItem)
            LogLevelMenuItem_ActionPerformed(event);
    }
}

void openMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    openMenuItem_ActionPerformed_Interaction1(event);
}

```

```

void aboutMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    aboutMenuItem_ActionPerformed_Interaction1(event);
}

void exitMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    exitMenuItem_ActionPerformed_Interaction1(event);
    exitMenuItem_ActionPerformed_Interaction2(event);
}

class SymComponent extends java.awt.event.ComponentAdapter
{
    public void componentShown(java.awt.event.ComponentEvent event)
    {
        Object object = event.getSource();
        if (object == domainLabel)
            domainLabel_ComponentShown(event);
    }
}

void inputsmenuitem_ActionPerformed(java.awt.event.ActionEvent event)
{
    inputsmenuitem_ActionPerformed_Interaction1(event);
}

void processingmenuitem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    processingmenuitem_ActionPerformed_Interaction1(event);
}

void updateStatLabels()
{
    numCheckLabel.setText(Stats._numChecks + " ");
    errorLabel.setText(Stats._notFound + " ");
    OCCountLabel.setText(Stats._numOutFiles + " ");
    fileOpenedLabel.setText(Stats._numInFiles + " ");
    bytesReadLabel.setText(Stats._bytesRead + " ");
    bytesOutputLabel.setText(Stats._bytesWritten + " ");
    domainLabel.setText(Config.getDomain());
    configLabel.setText(Config.getFile());
    inFileLabel.setText(Config.getCurInFile());
    outFileLabel.setText(Config.getCurOutFile());
    errorLabel.setText(Stats._notFound + " ");
    dateLabel.setText(Config.getDate());
}

void schedmenuitem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    schedmenuitem_ActionPerformed_Interaction1(event);
}

void exitMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    if (Config.modified())
        try {
            // ConfirmSaveExitDialog Create and show as modal

```

```

        (new ConfirmSaveExitDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

void genButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    _halt=false;
    _opt = new Optimizer(this);
    _opt.setFileList(Config.GetFiles());
    _opt.start();
    return;
}

void button4_ActionPerformed(java.awt.event.ActionEvent event)
{
    textAreal.setText("");
    Stats.clear();
    updateStatLabels();
    repaint();
}

void domainLabel_ComponentShown(java.awt.event.ComponentEvent event)
{
    domainLabel.setText(Config.getDomain() );
}

void saveMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
    saveMenuItem_ActionPerformed_Interaction1(event);
}

void saveMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        Config.save();
    } catch (Exception e) {
    }
}

void RmButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    _opt = new Optimizer(this);
    // _opt.start();

    OutMgr.print("-----> "
        + "Site cleanup started at "
        + new Date().toLocaleString()
        + " <-----" + "\n\n");
    OutMgr.print(" ");
    try {
        // ConfirmDialog Create and show as modal
        ConfirmDialog cfd = new ConfirmDialog(this, true);
        cfd.setVisible(true);
        if (cfd.deleteOK())
        {
            String[] flist = Config.GetFiles();
            if (flist != null)
            //TODO: handle case where only one file and no directory is specified in Config
            for (int i = 0; i< flist.length; i++)
            {
                File listin=new File(flist[i]);

                if ( !listin.exists() )
                OutMgr.print(" **** ERROR: Specified input file not found **** " +
                    OutMgr.chopPath(flist[i],maxplen) + " - skipping" );
            }
        }
    }
}

```

```

        else
        {
            if (listin.isDirectory() )
                _opt.deleteOCFiles(flist[i].trim(), null);
            else
            {
                _opt.deleteOCFiles(listin.getParent(), listin.getName() );
            }
        }
    }
}

catch (Exception e) {}
OutMgr.print(" ");
OutMgr.print("-----> " +
    "Site cleanup ended at " + new Date().toLocaleString()
    + " <-----" );

    _opt.stop();
    _opt.destroy();
}

synchronized void stopButt_ActionPerformed(java.awt.event.ActionEvent event)
{
    halt();
    notifyAll();
    OutMgr.print(" ");
    OutMgr.print("-----> Shutdown: finishing current input
<-----");
    OutMgr.closeLog();
}

void WWWhooshServer_windowClosing(java.awt.event.WindowEvent event)
{
    // to do: code goes here.
    WWWhooshServer_windowClosing_Interaction1(event);
    WWWhooshServer_windowClosing_Interaction2(event);
}

void WWWhooshServer_windowClosing_Interaction1(java.awt.event.WindowEvent event)
{
    if (Config.modified())
        try {
            // ConfirmSaveExitDialog Create and show as modal
            (new ConfirmSaveExitDialog(this, true)).setVisible(true);
        } catch (Exception e) {}
}

void exitMenuItem_ActionPerformed_Interaction2(java.awt.event.ActionEvent event)
{
    try {
        this.dispose();
        System.exit(0);
    } catch (Exception e) {}
}

void WWWhooshServer_windowClosing_Interaction2(java.awt.event.WindowEvent event)
{
    try {
        this.dispose();
        System.exit(0);
    }
}

```

```

        } catch (Exception e) {
        }
    }

    void saveAsMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
    {
        saveAsMenuItem_ActionPerformed_Interaction1(event);
    }

    void saveAsMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
    {
        FileDialog fileDialog1 = new java.awt.FileDialog((Frame)this);

        fileDialog1.setFile(Config.getFile() );
        fileDialog1.setTitle("WWWhoosh Specify Config Filename");
        fileDialog1.setVisible(true);
        String inFileName = fileDialog1.getFile();
        String dirname = fileDialog1.getDirectory();
        Config.setFile(inFileName);
        Config.setRootDir(dirname);
        Config.saveAs(dirname + inFileName);
        this.updateStatLabels();
    }

    void openMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
    {
        FileDialog fileDialog1 = new java.awt.FileDialog((Frame)this);
        //fileDialog1.setDirectory( Config.getCurDir() );
        fileDialog1.setFile(Config.getFile() );
        fileDialog1.setTitle("WWWhoosh Specify Config Filename");
        fileDialog1.setVisible(true);
        String inFileName = fileDialog1.getFile();
        String dirname = fileDialog1.getDirectory();
        Config.loadFrom(dirname + inFileName);
        Config.setFile(dirname + inFileName);

        this.updateStatLabels();
    }

    void logbutt_ActionPerformed(java.awt.event.ActionEvent event)
    {
        if (OutMgr.fileOut() )
        {
            logbutt.setBackground(Color.lightGray );
            OutMgr.print("Logfile output stopped at " +
                new Date().toLocaleString() );
            logbutt.setLabel("Logfile OFF");
            OutMgr.setLogOff();
        }
        else
        {
            OutMgr.setLogOn();
            //logbutt.setBackground( new Color( 102,153,153 ) );
            logbutt.setLabel("Logfile ON");
            OutMgr.print("Logfile output started at "
                + new Date().toLocaleString() + "\n" );
        }
    }

    void inputsmenuitem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
    {
        try {
            // InputsDialog Create and show as modal
            (new InputsDialog(this, true)).setVisible(true);
        } catch (Exception e) {
        }
    }

    void processingmenuitem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)

```



```
{
    try {
        // ProcessingDialog Create and show as modal
        (new ProcessingDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

void schedmenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // SchedulingDialog Create and show as modal
        (new SchedulingDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

void aboutMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // AboutServerDialog Create and show as modal
        (new AboutServerDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}

void LogLevelMenuItem_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.

    LogLevelMenuItem_ActionPerformed_Interaction1(event);
}

void LogLevelMenuItem_ActionPerformed_Interaction1(java.awt.event.ActionEvent event)
{
    try {
        // LogLevelDialog Create and show as modal
        (new LogLevelDialog(this, true)).setVisible(true);
    } catch (Exception e) {
    }
}
}
```

```

package WWWhooshCommon;

import COM.reitshamer.http.OCResponse;
import COM.reitshamer.http.OCDResponse;
import COM.reitshamer.util.*;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.util.zip.GZIPInputStream;
import java.net.Socket;
import java.net.InetAddress;
import java.util.Hashtable;
import java.util.Date;

/***** OCCACHE *****/
//
// This class is responsible for maintaining the OC information
// at runtime for the proxy.
// The request-handling logic consults OCCache to keep track
// of timestamps, OC files requested but not found, and OC/OCD files
// already retrieved.
//
/***** OCCACHE *****/
public class OCCache
{
    //private String _etag;
    private String _domain = "none";
    private String _thisfileURL="";
    private Hashtable _sockets = new Hashtable(); //sockets keyed by InetAddress
    private Hashtable _checks = new Hashtable(); //URL and timestamp per subobject
    private Hashtable _notfound = new Hashtable(); //URL and "NF" if previously not found
    private Hashtable _processed = new Hashtable(); //URL and date if OC has been fetched
    private Hashtable _OCDCache = new Hashtable(); //OCD file URL and int timestamp or -1 if
    fetch-in-progress
    private boolean _building = false;

    private static long socket_create_total_ms=0;
    private static long socket_count=0;
    private static Socket _cachedSocket=null;

    //record version number of
    public int getVersionNumber()
    {
        return 0;
    }

    /***** *****/
    public void clearCaches()
    {
        _checks.clear();
        _OCDCache.clear();
        _processed.clear();
        Debug.print(2, " OC and OCD caches cleared ");
    }

    /***** *****/
    public Socket getSocket(String host, int port ) throws
        java.net.UnknownHostException, java.io.IOException
    {
        if (Debug.useProxy)
            return new Socket(Debug.outAddress, Debug.outProxy);
        else
            return new Socket(host, port);
    }

    /***** *****/

```

```

public String getthisFile()
{return _thisfileURL; }

/*****/
//set flag to indicate directory is building to GET threads
public synchronized void setBuilding(boolean b)
{ _building=b;}
public synchronized boolean getBuilding()
{return _building;}

/*****/
//keep track of non-existent and already-fetched OCD files to avoid second GETs
//TODO: add eviction behavior (to whole cache, actually)
public void putProcessed(String s, Date d)
{ _processed.put(s,d);
  Debug.print(2, "    remember fetched: " + Debug.chopPath(s) );
}

/*****/
public Date getProcessed(String s)
{
  Date tmp = (Date) _processed.get(s);
  return tmp;
}

/*****/
//keep track of non-existent and already-fetched OCD files to avoid second GETs
public void putNotFound(String s)
{ _notfound.put(s,"NF");
  Debug.print(2, "    remember OC NotFound: " + Debug.chopPath(s) );}

/*****/
public Object getNotFound(String s)
{Object tmp = _notfound.get(s);
  if (tmp != null)
    Debug.print(2, Debug.chopPath(s) + "    previously not found, skip GET \n");
  return tmp;}

/*****/
public int size()
{return _checks.size();}

/*****/
public void clearChecks()
{ _checks.clear();}

/*****/
public byte[] uncompressBLOB(byte[] blob)
{
  ByteArrayInputStream bais = new ByteArrayInputStream(blob);
  ByteArrayOutputStream uncompressedstream = new ByteArrayOutputStream();
  byte [] fixbuf = new byte[blob.length];
  try
  {
    GZIPInputStream gzis = new GZIPInputStream(bais);
    int fullcount= gzis.read(fixbuf) ;
    while (fullcount != -1)
    {
      uncompressedstream.write(fixbuf,0,fullcount);
      fullcount= gzis.read(fixbuf);
    }
  }
  catch (IOException ioe)
  {
    Debug.forceprint(" **** ERROR uncompressing binary object \n" );
  }

  return  uncompressedstream.toByteArray();
}

```

```

/*****/
public synchronized void putBLOB(String fname, byte[] blob)
{
    int cut = fname.indexOf("/./");
    while ( cut != -1)
    {
        String part1 = fname.substring(0,cut+1);
        String part2 = fname.substring(cut+3);
        fname = part1 + part2;
        cut = fname.indexOf("/./");
    }

    cut = fname.indexOf("/../");
    while ( cut != -1)
    {
        String part1 = fname.substring(0,cut+1);
        String part2 = fname.substring(cut+4);
        fname = part1 + part2;
        cut = fname.indexOf("/../");
    }

    //TODO: use a more foolproof stub indicator than zero!
    if (blob[0] == 00)
        Debug.print(2,"  cache entry: " + Debug.chopPath(fname) );
    else
        if (blob[0] == 0x1F)
        {
            Debug.print(1,"  uncompressing from: " + blob.length );
            blob = uncompressBLOB(blob);
            Debug.print(1, " to fullsize: " + blob.length );
        }
        else
            Debug.print(2,"  caching object: " + Debug.chopPath(fname) );

    _OCDCache.put(fname, blob);

    //wake up threads waiting to read  blobs
    this.notifyAll();
}

/*****/
//thread-synched method to retrieve binary data for a SRC object
//will wait if OCD directory build is underway - will wait if fetch
//of named object is underway.
public synchronized byte[] getBLOB(String fname)
{
    String lowstr = fname.toLowerCase();//clamp to lowercase for matching purposes

    if ( getDate(fname) == null )        //if there is no OC entry for this file
        return null;

    if ( _building)
        Debug.print(1,"  directory wait: GET " + Debug.chopPath(lowstr) );

    while ( _building)
        try { wait(); }
        catch (InterruptedException iex)
        {}

    byte tmp[] = (byte[])_OCDCache.get((String)lowstr);
    if (tmp == null)
        return null;

    if (tmp[0] == 00)
        Debug.print(1," BLOB wait: GET " + Debug.chopPath(lowstr) );

    while (tmp[0] == 00 )
        {try {wait(); }
        catch (InterruptedException iex)

```

```

    {}
    tmp = (byte[])_OCDCache.get((String)lowstr);
}

if (tmp == null)
    Debug.print(2, " BLOB notfound: " + Debug.chopPath(lowstr) );

return tmp ;
}

/*****/
public synchronized void clearOCD()
{
    _OCDCache.clear();
    _building=false;
    notifyAll();
}

/*****/
//methods to add a timestamp/url pair, remove it, and key it
public void putCheck(String s, Date d)
{
    s = s.toLowerCase();
    int cut = s.indexOf("/./");
    while ( cut != -1)
    {
        String part1 = s.substring(0,cut+1);
        String part2 = s.substring(cut+3);
        s = part1 + part2;
        cut = s.indexOf("/./");
    }

    cut = s.indexOf("/../");
    while ( cut != -1)
    {
        String part1 = s.substring(0,cut+1);
        String part2 = s.substring(cut+4);
        s = part1 + part2;
        cut = s.indexOf("/../");
    }

    //TODO: handle mixed case
    Debug.print(2, "stamp: " + Debug.chopPath(s) );
    _checks.put(s,d);
}

/*****/
public void removeCheck(String s)
{
    _checks.remove(s);
}

public Date getDate(String s)
{
    String lowers = s.toLowerCase();

    Date tmp = (Date)_checks.get(lowers);
    if (tmp == null)
        Debug.print(1, " no stamp for " + Debug.chopPath(lowers));
    return tmp;
}

/*****/
//bunch of rules to decide whether this URI is an HTML file
//based on extension or its being the default file
public boolean isHTMLExt(String uri)
{
    //TODO: replace this check to allow any extension from server config
    int extdot = uri.lastIndexOf(".");
    if (extdot > 0)
    {
        String ext = uri.substring(extdot);
    }
}

```

```

        if (ext.startsWith(".htm"))
            return true;
        if (ext.startsWith(".asp"))
            return true;

        return false;
    }

    //detect the "noname" case
    if ( ( uri.length()==0 ) || (uri.equals("/") ) ||
        (uri.endsWith("/") && uri.indexOf(".") == -1) )
        return true;

    return false;
}

/****/
//Decides: Should we find the OCD file??
public boolean needsOCD(String host, String uri, String orig_timestamp)
{
    // Logic:
    // If this is an HTML file GET, AND
    // If there's no timestamp,
    // AND an OCD format exists for this HTML file,
    // AND the OCD format itself has not yet been fetched
    // return TRUE, else FALSE
    // Case: default URL to HTML file specified
    if (!isHTMLExt(uri)) //ignore non-HTML files
        return false;

    String OCD_url = host + prefixURI("ocd_", uri);

    if ( getProcessed(OCD_url) != null )
        return false;

    if (orig_timestamp != null)
        return false;

    //for noname files, always get OCD
    //TODO: bug: translate to default filename
    Date OCD_info_available = (Date)getDate(OCD_url);
    Date OCD_info_fetched = (Date)_OCDCache.get(OCD_url);

    if ((OCD_url.indexOf("noname") != -1) && (OCD_info_fetched == null) )
        return true;

    //OCD file is named in OC cache but not yet in OCD cache
    if ((OCD_info_available != null) && (OCD_info_fetched == null) )
        return true;

    return false;
}

/****/
//Utility method -
//normalize URI - if it needs an ending slash, add one.
//if there is no ending filename, extend by asking for "noname.htm"
//the URI is then converted to an OC_request
//
public String prefixURI(String prefix, String uri)
{
    String newuri = uri.toLowerCase(); //clamp everything to lower case to help matching
    newuri = newuri.replace('\\', '/'); //normalize slash direction

    int firstslash = uri.indexOf("/");
    int lastslash = uri.lastIndexOf("/");
    int lastdot = uri.lastIndexOf("."); //find extension, if there is one

    String rest = "";

```

```

if ((firstslash == 0) && (lastdot > lastslash)) //at least a slash and a dot after
    rest = uri.substring(lastslash+1); //take all after last slash

if ( ( uri.length()==0 ) || (uri.equals("/") ) ||
    (uri.endsWith("/") && (uri.indexOf(".") == -1) )
    )//if there's no slash, or nothing after it
    rest = rest + "noname.htm"; //look for the noname file.

//A problem form:
//www.foo.com/adfa.cl?aa=bb
//we need to distinguish this from
if ((rest.indexOf(".htm") != -1 ) || (rest.indexOf(".asp") != -1 ) )
    //we have an HTML file after normalizing
    //Note: we need to recognize other extensions
{
    //synthesize OC_ filename from given URI
    rest = prefix + rest;

    if (uri.endsWith("/") && (lastdot > 0)) // dot, slash
        newuri = uri.substring(0,lastslash+1) + rest;

    if (!uri.endsWith("/") && (lastdot == -1)) //no dot, no slash
        newuri = uri + "/" + rest;

    if (!uri.endsWith("/") && (lastdot > 0)) // dot, no slash
        newuri = uri.substring(0,lastslash+1) + rest;

    if (uri.endsWith("/") && (lastdot == -1)) // no dot, slash
        newuri = uri.substring(0,lastslash+1) + rest;
}

newuri = newuri.replace('\\', '/'); // clean up those bloody slashes!!!
return newuri;
}

/*****
// Build the OC cache from an OC response by parsing its string
//
public synchronized void buildFrom(OCResponse OCR)
{
    String checks = OCR.getContentString();
    //Record date and tag for the OC file we just parsed
    //This will tell us if we can avoid fetching it again
    String OCdate = OCR.getHeader("Last-Modified");

    buildFromString(checks, OCdate);
}

//parse an input coherency string into the OC cache
//
public void buildFromString(String checks, String OCdate)
{
    String fname = null;
    String fdate = null;
    int sindx=0;

    int after=checks.indexOf("&",sindx);
    boolean firstfile = true;
    _thisfileURL="";

    while (after != -1)
    {
        fname = checks.substring(sindx,after).replace('\\', '/');

```

```

sindx=after+1;
after=checks.indexOf(",",sindx);
fdate=checks.substring(sindx, after).trim();
Date d = null;

if (fdate.startsWith("@") ) //get date from timestamp as long
{
    Long dstamp = new Long(fdate.substring(1));
    d = new Date(dstamp.longValue());
}
else
{d = new Date(fdate);}

sindx=after+1;
after=checks.indexOf("&",sindx);
this.putCheck(fname.trim(), d);

if (firstfile)
{
    //in case this is the OC_noname, record what noname should map to
    _thisfileURL=fname.trim();
    firstfile=false;
}
}

if (OCdate != null)
{
    Date Octime = new Date(OCdate);
    this.putCheck(fname, Octime);
}

Debug.print(2, "    cached total of: " + this.size() + " checks. " );
Counters.afterBuild();
}

}

/***** HEADER *****/
//
// Encapsulates the header format of OC and OCD files
//
//
package WWWhooshCommon;

import java.io.*;
import Build;
import Config;
import OutMgr;
import java.util.Date;

public class WhooshHeader
{
    private long _datapointer=0;
    private short _version = 0;
    private short _build = 0;
    private int _serverID = 0;
    private long _serverKey = 0;
    private long _gentime = 0;
    private short _subcount = 0;
    private int _formats = 0;
    private int _proc_tag = 0;
    String _h1 = "WWWhoosh (c) MTI \r\n";
    String _h2 = "Patents Pending \r\n";
    String _h3 = "Version 3.1 \r\n";
    String _h4 = "12345-56789 \r\n";
    String _h5 = new Date(Config.getRuntime()).toGMTString() + "\r\n";

    public int readFrom(BufferedInputStream xis) throws IOException
    {

```



```

//read header and initialized based on version, etc.
//Otherwise assume we have prototype-era files and just begin processing
DataInputStream headerstream = new DataInputStream(xis);
int byteread=0;

//TODO: do this more efficiently if needed
while ( (byteread != -1) && (byteread != (int)26) )
    byteread = xis.read();

_datapointer = headerstream.readLong();
_version     = headerstream.readShort();
_build       = headerstream.readShort();
_serverID    = headerstream.readInt();
_serverKey   = headerstream.readLong();
_gentime     = headerstream.readLong();
_subcount    = headerstream.readShort();
_formats     = headerstream.readInt();
_proc_tag    = headerstream.readInt();
return 142; //TODO: hack! record actual size of header read...
}

//write the header portion of an WWWhoosh file to the output stream
public void writeTo( FileOutputStream fos )
{
    DataOutputStream dos= new DataOutputStream(fos);

    short version = (short)Build.VERSION;    //WWWwhoosh Version
    short build   = (short)Build.NUMBER;      //Server BUILD number
    int  serverid = 134134;                   //TODO: how do we fetch a valid server ID?
    long licensekey = 111111111;             //TODO: generate valid license key
    short subobjectcount=0;                  //number of subobjects in base HTML page
    int  availableFormats=1;                 //bit per available OCD file
    int  processingflags=0;

    if (Config.getGZip())//refers to and describes coherency format
        processingflags=1;

    int  datapointer=42;                    //self-relative pointer to data

    try
    {
        dos.write(_h1.getBytes());
        dos.write(_h2.getBytes());
        dos.write(_h3.getBytes());
        dos.write(_h4.getBytes());
        dos.write(_h5.getBytes());
        //TODO - write ctrl-z

        dos.writeByte(26);                  //begin machine-readable portion
        dos.writeLong(datapointer);          //8
        dos.writeShort(version);             //2
        dos.writeShort(build);               //2
        dos.writeInt(serverid);              //4
        dos.writeLong(licensekey);           //8
        dos.writeLong(Config.getRuntime());  //8 //generation timestamp
        dos.writeShort(subobjectcount);      //22
        dos.writeInt(availableFormats);      //4
        dos.writeInt(processingflags);       //4

        //After this Header comes the "content" part of each OC file - e.g coherency
        //or subobject payload
    }
    catch (IOException ioe)
    {
        OutMgr.print(" **** ERROR: Exception writing file header **** ");
    }
}

public short getVersion()
{

```

```
    return _version;
}

public boolean compression()
{
    //TODO: use bitmask
    if (_proc_tag == 1)
        return true;
    return false;
}
}
```

What is claimed is:

1           1. The method for processing and transmitting a hypertext data object which  
2 comprises, in combination, the steps of:  
3           at an Internet server location:  
4                 scanning the content of a given hypertext data object to identify  
5                 at least one external data object that is referred to in said hypertext data  
6                 object,  
7                 combining said given hypertext data object and said external  
8                 data object to form a single compound object containing data from said  
9                 hypertext data object and said external data object, and  
10                assigning a unique resource name to said compound object  
11                which is derived from the resource name of said given hypertext data  
12                object,  
13           at an Internet client location:  
14                deriving said unique resource name from said resource name of  
15                said given hypertext data object,  
16                transmitting a request for said single compound data object  
17                specified by said unique resource name to said server location,  
18                receiving said single compound data object when from said  
19                server location in response to said request, and  
20                processing said compound data object to render said hypertext  
21                data object and said external data object at said client location,  
22           whereby the data needed to render said hypertext data object is obtained in a  
23 single request-response transaction rather than a plurality of request-response  
24 transactions.

1           2. The method of avoiding unnecessary network request-response exchanges  
2 when valid copies of information included by reference in a hypertext data object is  
3 already stored in the local cache at a client location, said method comprising, in  
4 combination, the steps of:  
5           at an Internet server location:

6                   analyzing the content of a given hypertext data object to  
7                   identify at least one external data object that is referred to in said  
8                   hypertext data object,  
9                   forming a single compound data object which contains  
10                  information describing said given hypertext data object and  
11                  information describing the time at when said external data object was  
12                  last modified at said server location, and  
13                  transmitting said single compound data object from said server  
14                  location to said client location, and  
15                  at said client location,  
16                  receiving said compound data object from said server location,  
17                  and  
18                  comparing said information describing the time when said  
19                  external data object was last modified at said server location with the  
20                  time when a copy of said external data object was last stored in said  
21                  local cache.